INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

ProQuest Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

STOCHASTIC OPTIMIZATION MODELS FOR SERVICE AND MANUFACTURING INDUSTRY

BRIAN T. DENTON, B.Sc., M.Sc.

A Thesis

Submitted to the School of Graduate Studies

in Partial Fulfilment of the Requirements

for the Degree

Doctor of Philosophy

McMaster University

STOCHASTIC OPTIMIZATION MODELS

Doctor of Philosophy (2001) McMaster University

Management Science Hamilton, Ontario

TITLE: Stochastic Optimization Models for Service and Manufacturing Industry

AUTHOR: Brian T. Denton, B.Sc (McMaster University), M.Sc. (York University)

SUPERVISOR: Professor Diwakar Gupta

NUMBER OF PAGES: xi, 156

Abstract

We explore two novel applications of stochastic optimization inspired by real-world problems. The *first* application involves the optimization of appointments-based service systems. The problem here is to determine an optimal schedule of start times for jobs that have random durations, and a range of potential cost structures based on common performance metrics such as *customer waiting* and *server idling*. We show that the problem can be formulated as a two-stage stochastic linear program and develop an algorithm that utilizes the problem structure to obtain a near-optimal solution. Various aspects of the problem are considered, including the effects of job sequence, dependence on cost parameters, and job duration distributions. A range of numerical experiments is provided and some resulting insights are summarized. Some simple heuristics are proposed, based on relaxations of the problem, and evidence of their effectiveness is provided. The *second* application relates to inventory deployment at an integrated steel manufacturer (ISM). The models presented in this case were developed for making inventory *design-choice* (what to carry) and *lot-size* (how much to carry) decisions. They were developed by working with managers from several different functional areas at a particular ISM. They are, however, applicable to other ISMs and to other continuous-process industries with similar architectures. We discuss details of the practical implementation of the models, the structure of the problems, and algorithms and heuristics for solving them. Numerical experiments illustrate the accuracy of the heuristics, and examples based on empirical data from an ISM show the advantages of using such models in practice and suggest some managerial insights.

Acknowledgments

This research was supported in part by the National Science Foundation (DMII-9988721), the Natural Sciences and Engineering Research Council of Canada. Research in Chapters 4 and 5 was also supported by Dofasco Inc. of Hamilton, Ontario, Canada through research grants. Several people from Dofasco deserve thanks. Keith Jawahir layed the groundwork for the slab inventory deployment project initiative, and helped make several important issues clear to me through many useful discussions. I am also grateful to many enthusiastic participants in this project, including Tom Braun, Maureen Crane, Robin Hallam, Ron Johnson, Mike Moore, and Jane Wood. I especially owe thanks to my doctoral supervisor, Diwakar Gupta, for many helpful discussions about research topics, guidance in developing the topics explored, and timely feedback during development and preparation of the thesis.

I also want to thank my family and friends for the endless moral support I received during my doctoral studies. My parents helped me to develop an attitude and commitment to work without which I certainly would not have completed my doctorate. Thanks to my friends at McMaster (Kaywan, Laurent, Paul, Spiro, and Mustafa) for our Friday (especially kirok) trips to the Phoenix to argue and gloat around the shuffle board, softball (if you can call it that), and all the summer barbecues. Most of all I want to thank my wife Laura for her constant love and support, not complaining about me coming home late and working weekends, and pushing me to finish my thesis on time even when it sometimes seemed like it would never end.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Both manufacturing and service sector industries are significantly affected by supply and demand uncertainty. In the manufacturing sector (e.g. automobile assembly, steel-making, semiconductor fabrication) uncertainty in production yield, processing time, and customer demand, affects long-term decisions such as capacity investment and facility location and design, as well as medium-range decisions involving inventory control and production planning. Service sector industries (e.g. hospitals, banks, law practices) are similarly affected. For them the major source of supply uncertainty is often randomness in service time durations, which affects performance measures such as facility idle time, customer waiting and overtime costs. These measures in turn affect long-term decisions such as capacity investment and service pricing, as well as short-term decisions about daily staff assignment and work/shift schedules.

An important aspect of managing uncertainty is understanding the short-term recourse actions that can be taken after uncertain future events are realized. For instance, there may be significant variation over time in customer demand for a product or service but the overall optimal capacity may be far short of the peak demand. As a result, different types of recourse available for managing congestion should be considered when making a capacity investment decision. For example, a higher peak load pricing is used in many

service industries to encourage customer utilization at times other than the peak load time, and in manufacturing industries planned overtime and outsourcing of product orders are common.

Effective stochastic planning models must be able to capture both the relevant sources of uncertainty as well as the potential recourse actions. In many cases, identifying sources of uncertainty that affect system performance is easier than modeling them. The modeler must be able to ascertain properties of, and associations between, the random variables (e.g. moments, correlation coefficients). In most *real world* applications this is done by analyzing historical data and/or obtaining subjective information from decision makers. To model the recourse decisions adequately the modeler must have a good understanding of the different ways the system can react to potential realizations of uncertain future events.

Mathematical models for decision making under uncertainty originated in statistical decision theory, developed initially be Wald (1950). Such models focus on procedures for constructing objectives and updating the probability distributions of random variables, based on partial decisions and observations. The stochastic optimization literature, on the other hand, focuses on the use of mathematical programming models. The first such models, advanced by Beale (1955) and Dantzig (1955), dealt with two-stage problems in which there is an initial decision, followed by the resolution of some uncertainty, and then subsequent recourse actions. They were deterministic linear programs in which the objective was an expectation with respect to a discrete set of *scenarios*. An alternative formulation, called *chance constrained programming*, was developed by Charnes and Cooper (1959), and includes constraints that hold with some specified probability. Two-stage modeling is readily generalized to multi-stage modeling which can be considered a branch of discrete-time linear control. In contrast to statistical decision theory, the emphasis of research on these mathematical programming-based models has been on solution methods and analytic properties of solutions.

A general stochastic decision problem can be expressed in the following way. Let $\mathbf{x}$ represent some decision vector and $\boldsymbol{\xi}$ a vector of random variables with support Ξ and probability distribution function P. Assuming $r(\mathbf{x}, \cdot) : \Xi \mapsto \Re$ represents some reward we can write a general decision model as

$$\max_{\mathbf{x}} \left\{ E_{\boldsymbol{\xi}}[r(\mathbf{x}, \boldsymbol{\xi})] \mid \mathbf{x} \in X \right\} \tag{1.1}$$

where

$$E_{\boldsymbol{\xi}}[r(\mathbf{x}, \boldsymbol{\xi})] = \int_{\Xi} r(\mathbf{x}, \boldsymbol{\xi}) dP(\boldsymbol{\xi}) \tag{1.2}$$

and X is the set of feasible decisions. The difference between stochastic programming and statistical decision theory is the approach to this problem. Statistical decision theory is concerned with the appropriate method for updating P as partial information is obtained as partial choices of decision variables $\mathbf{x}$, and observations of $\boldsymbol{\xi}$. This approach is typically tractable only when the feasible decisions, X, form a small, finite set. On the other hand, in stochastic programming it is assumed that the form of the function $r(\mathbf{x}, \boldsymbol{\xi})$ and changes in P, as a function of the chosen decisions, are known. The difficulty is assumed to be in evaluating the expectation in (1.1), which in most realistic decision models requires the evaluation of multi-dimensional integrals (or nested sums if the random variables are discrete) with no closed-form solutions. The difficulty in evaluating the objective and gradient of (1.1) often makes standard optimization methods impractical. However, there are several methods that are specifically designed for problems of this type. We review some of them in the next Chapter.

In recent years, much of the focus in the literature on stochastic programming has shifted to identifying properties of specific problems that allow for efficient solution or approximation. In this thesis we explore two novel stochastic optimization models that are inspired by real-world problems. The first is in the determination of the optimal schedule of start times for jobs that have random durations, and a range of potential cost structures

based on common performance metrics such as *customer waiting* and *server idling*. That problem can be described as optimization of the performance metrics of an $S(n)/G(n)/1$ queue, where $S(n)$ stands for a set of n scheduled arrivals, and $G(n)$ the job duration probability distributions for each of the n jobs. We show that the problem can be formulated as a two-stage stochastic linear program and develop an algorithm that utilizes the problem structure to obtain a near-optimal solution, as well as upper and lower bounds on the difference between the optimal and approximate solution. Bounds on the expected cost savings that can be realized from resequencing jobs are derived. Since variability in job durations is a major source of inefficiency in appointment scheduling, we also study the effect of changing the variance of job durations on the total expected cost of running an appointments-based service system. A range of numerical experiments for different cost structures and job duration distributions is presented and general insights gained from the numerical results are summarized. Some simple heuristics are proposed based on relaxations of the problem, and evidence of their effectiveness is provided.

The second application relates to inventory planning at an integrated steel manufacturer (ISM). Although ISMs have typically operated according to a *make-to-order* policy in the past, recent and significant changes to the competitive environment in which they operate have resulted in the need to carry planned inventory. Existing inventory models were developed with applications to discrete parts manufacturing in mind and generally are not applicable to the inventory problems faced in process industries like the steel industry. The models we present can be used for making strategic *inventory deployment* decisions which consist of choosing the design of inventory items to carry and the quantity to carry. They can easily be extended to other continuous-process industries with similar architectures. First, we begin by presenting a model for choosing, from a continuous range of semi-processed inventory, a finite set of items to stock. The model accounts for the large variety of finished products, limited storage capacity, and production costs and constraints. The problem is

classified with respect to the existing literature on similar problems and known heuristics are adapted for solving it. Details of the practical implementation of the model at a particular ISM are discussed. Numerical experiments based on empirical data are presented and the managerial insights that can be drawn from them are addressed. Next, we discuss a related model for simultaneously choosing items to stock and stocking levels, given uncertainty in both supply and demand. The structure of the problem is analyzed, and heuristics that are well suited to the problem are discussed. Their accuracy is tested in a series of numerical experiments, and several numerical examples based on empirical data are presented which illustrate the importance of explicitly modeling uncertainty for inventory planning.

The thesis is organized as follows. Chapter 2 presents a literature review of stochastic programming methods and applications. In Chapter 3 we introduce the problem of scheduling start times for jobs with uncertain durations, discuss the model and methods used, discuss some heuristics, and provide several numerical examples as well as general insights. Chapter 4 introduces the steel-making process and discusses some important aspects of steel-making that affect optimization of inventory systems for semi-finished products; it then presents a model for optimizing inventory placement, methodology for solving it, numerical examples, and specific details of its implementation at a particular ISM. In Chapter 5 we consider the problem of simultaneously choosing the items to stock and the level of inventory to order given uncertainty affecting supply and demand. A stochastic linear programming model is presented, heuristics suited to the structure of the problem are discussed and tested, and numerical examples based on empirical data from the particular ISM are provided and used to draw managerial insights. Finally, in Chapter 6, we provide a summary and discuss future research directions.

Chapter 2

Literature Review

2.1 Introduction

Stochastic programming is the branch of mathematical programming concerned with solving optimization problems in which there is uncertainty in the problem data. Typically such problems are related to dynamic systems. They involve decisions that are made given only partial knowledge of future outcomes. The optimal decisions are defined by the probabilistic nature of the future outcomes and the potential corrective (recourse) actions which can be taken after they are realized. Modeling of such problems under uncertainty can lead to a wide range of problem structures. In this Chapter we review the structures and properties of stochastic linear programs (SLP) and methods for solving them.

The key feature that distinguishes SLPs from linear programs is the presence of random variables that affect the problem parameters. We denote the realizations of such random variables which affect the problem parameters by the vector $\xi(\omega) \in \Re^n$, where ω is a random object defined on some abstract probability space $(\Omega, \mathcal{A}, P)$, such that Ω is the set of possible outcomes, $\mathcal{A}$ is the set of events, and P is a probability measure. The vector $\xi(\omega) \in \Re^n$ is a mapping of the abstract probability space into $(\Re^n, \mathcal{B}^n, F)$ where $\mathcal{B}^n$ is the Borel field on $\Re^n$ and F is the distribution function. Furthermore, $\Xi \subset \Re^n$ denotes the

support of ξ, i.e., the smallest closed set in $\Re^n$ such that $P(\xi(\omega) \in \Xi) = 1$. Throughout this Chapter we write vectors in bold face, use a bar (e.g. $\bar{\xi}$) to denote first moments, and suppress notation for the transpose when writing inner products of vectors except when it is not obvious from the context.

We start by using a simple example of a stochastic model, the *newsvendor problem*, to illustrate some properties of stochastic programs. The problem involves a decision about how many units of a product (newspapers) to purchase for resale, given random demand. The decision, $x \in \Re$, is the quantity of papers to purchase. The per unit cost of shortages is c_s, and the per unit cost of excess papers is c_e, and demand is a random variable, ξ, with probability distribution function $F(\cdot)$. Thus we can write the cost function as

$$Q(x) = c_s \int_x^{\infty} (\xi - x) dF(\xi) + c_e \int_0^x (x - \xi) dF(\xi) \tag{2.1}$$

Under reasonable assumptions it can be shown that $Q(x)$ is convex and differentiable, and the optimality condition is as follows:

$$x^* = F^{-1}(\frac{c_s}{c_s + c_e}). \tag{2.2}$$

A common way of approximating stochastic models is to replace the random variables by their mean values. The resulting solution is called the *mean-value solution*. Using the mean-value approach in the newsvendor problem would result in the purchase of $\bar{\xi}$ units. Unless this happens to correspond to the optimality condition (2.1), i.e., $\bar{\xi} = F^{-1}(\frac{c_s}{c_s+c_e})$, the expected return is suboptimal. The difference between the expected return from a decision based on the mean-value approach and the true stochastic decision model is referred to as the *value of the stochastic solution* VSS (Birge, 1982). It is often used as a measure of the improvement due to solving the stochastic model. Another measure is the *expected value of perfect information* EVPI (Raiffa, 1961) which compares the solution to that which would be obtained if the decision maker knew the future outcome in advance. For the newsvendor

problem, if perfect information were available the expected cost would be 0 and the EVPI would be $-Q(x^*)$.

In the newsvendor example the analytic result (2.2) is easily derived. This is quite rare in many decision models as we will see from the more general setting of two-stage and multi-stage problems. In the sections to follow we review methods for solving general stochastic linear programs. The structure of the review is as follows. We start by discussing various formulations. Next we summarize some basic properties that underpin standard solution methods. The discussion of solution methods is partitioned into two groups: (a) exact algorithms for the case in which the support, Ξ, is finite and (b) sampling and bounding based approximations for the case in which the cardinality of Ξ is too large for exact solution, or when Ξ is continuous. For a more comprehensive discussion of stochastic programming the reader is referred to Kall and Wallace (1994), Dempster (1986), Ermoliev and Wets (1988), Frauendorfer (1992), Infanger (1994), Birge (1995) and Birge and Louveaux (1997).

This Chapter is organized as follows. In the next section we discuss the general formulation of two-stage and multi-stage stochastic linear programs, and the alternative *chance-constrained* formulation. In section 2.3 we summarize some basic properties of stochastic linear programs important to the development of practical solution methods. In section 2.4 we review some exact solution methods for the special case in which the support of the random variables is finite, and in section 2.5 we discuss some common approximations for the general case in which it may be continuous or finite. In section 2.6 we give some examples of applications of stochastic linear programming and in section 2.7 we briefly summarize some important future research directions.

2.2 Formulation of Stochastic Programs

In this section we briefly discuss the basic formulation of two-stage, multi-stage, and chance-constrained stochastic linear programs as well as an alternate formulation of such problems based on a redefinition of decision variables.

2.2.1 Two-Stage Stochastic Linear Programs (2S-SLPs)

Two-stage stochastic linear programs (2S-SLP) are used when there is a decision that must be made prior to the realization of some uncertain data, and linear functions provide a suitable approximation. After the uncertainty is resolved there is an opportunity for recourse. The recourse decision is defined by a second stage linear program which depends on the uncertain outcome and the first stage decision. The goal is to determine the first stage decision which best hedges against the different possible outcomes of the random variables, given that the optimal recourse action will be taken in the second stage. Letting $\mathbf{x} \in \Re^{n_1}$, and $\mathbf{y} \in \Re^{n_2}$, denote the first and second stage decisions respectively, we can write the general formulation of 2S-SLP as

$$\min_{\mathbf{x}} \{\mathbf{c}\mathbf{x} + Q(\mathbf{x}) \mid A\mathbf{x} = b\} \tag{2.3}$$

where $A \in \Re^{m_1 \times n_1}$, $\mathbf{b} \in \Re^{m_1}$, define deterministic constraints on $\mathbf{x}$ and $\mathbf{c} \in \Re^{n_1}$ is a vector of deterministic first stage cost coefficients. The recourse function, $Q(\mathbf{x})$, is

$$Q(\mathbf{x}) = E_{\boldsymbol{\xi}}[Q(\mathbf{x}, \boldsymbol{\xi})] = \int_{\Xi} Q(\mathbf{x}, \boldsymbol{\xi}) dP(\boldsymbol{\xi})$$

where

$$Q(\mathbf{x}, \boldsymbol{\xi}) = \min_{\mathbf{y}(\omega)} \{\mathbf{q}(\omega)\mathbf{y}(\omega) \mid \mathcal{T}(\omega)\mathbf{x} + \mathcal{W}(\omega)\mathbf{y}(\omega) = \mathbf{h}(\omega), \mathbf{y}(\omega) \geq 0\} \tag{2.4}$$

and $\mathbf{h} \in \Re^{m_2}$, $\mathcal{T} \in \Re^{m_2 \times n_1}$ and $\mathcal{W} \in \Re^{m_2 \times n_2}$. The second stage objective function, $Q(\mathbf{x}, \boldsymbol{\xi})$, and constraints are defined by a vector of cost coefficients, $\mathbf{q}(\omega)$, a vector of RHS values, $\mathbf{h}(\omega)$, and constraint matrices, $\mathcal{T}(\omega)$, and $\mathcal{W}(\omega)$, referred to as the *technology* and *recourse*

matrices respectively. The vector $\boldsymbol{\xi}$ may be comprised of some or all of the components of $\mathbf{q}(\omega)$, $\mathbf{h}(\omega)$, and the matrices $\mathcal{T}(\omega)$, and $\mathcal{W}(\omega)$, depending on the nature of the problem. In the most general case $\boldsymbol{\xi}(\omega) = (\mathbf{q}(\omega), \mathbf{h}(\omega), \mathcal{T}_{\cdot 1}(\omega), ..., \mathcal{T}_{\cdot n_1}(\omega), \mathcal{W}_{\cdot 1}(\omega), ... \mathcal{W}_{\cdot n_2}(\omega))$.

2.2.2 Multi-Stage Stochastic Linear Programs (MS-SLPs)

The 2S-SLP is a special case of the more general multi-stage stochastic linear program (MS-SLP) in which a partial resolution of uncertainty occurs at each of a discrete set of stages. At each stage there is a set of recourse actions that depend on prior decisions and realizations of random variables. The general MS-SLP with fixed recourse can be written as

$$\min\{\mathbf{c}^1\mathbf{x}^1 + E_{\boldsymbol{\xi}^2}[\min\{\mathbf{c}^2(\omega)\mathbf{x}^2(\omega^2) + ... + E_{\boldsymbol{\xi}^H}[\min\{(\mathbf{c}^H(\omega))\mathbf{x}^H(\omega^H)\}]\cdots\}]\} \tag{2.5}$$

$$\begin{aligned} s.t. \qquad \mathcal{W}^1\mathbf{x}^1 &= \mathbf{h}^1 \\ \mathcal{T}^1(\omega)\mathbf{x}^1 + \mathcal{W}^2(\omega)\mathbf{x}^2(\omega^2) &= \mathbf{h}^2(\omega) \\ \vdots \qquad &\vdots \\ \mathcal{T}^{H-1}(\omega)\mathbf{x}^{H-1}(\omega^{H-1}) + \mathcal{W}^H(\omega)\mathbf{x}^H(\omega^H) &= \mathbf{h}^H(\omega) \end{aligned}$$

$$\mathbf{x}^1 \geq 0, \quad \mathbf{x}^t(\omega^t) \geq 0, \quad t = 2, ..., H$$

where $t = 1, ..., H$ indexes H stages, and $\mathbf{x}^t \in \Re^{n_t}$, $\mathbf{h}^t \in \Re^{m_t}$, $\mathcal{T}^t \in \Re^{m_t \times n_t}$ and $\mathcal{W}^t(\omega) \in \Re^{m_t \times n_t}$. The cost vector, $\mathbf{c}^1 \in \Re^{n_1}$, is known in stage 1 and the constraints $\mathcal{W}^1\mathbf{x}^1 = \mathbf{h}^1$ correspond to the first stage constraints from the two-stage formulation (2.3). Random observations made at the beginning of stage $t+1$ are represented by ω_t and the history of observations made up to stage $t+1$ are represented as $\omega^t = \{\omega_1, ..., \omega_t\}$. The random observations that are made during stage t are represented by

$$\boldsymbol{\xi}^t = (\mathbf{c}^t(\omega), \mathbf{h}^t(\omega), \mathcal{T}_{\cdot 1}^{t-1}(\omega), ..., \mathcal{T}_{\cdot n_t}^{t-1}(\omega), \mathcal{W}_{\cdot 1}^t(\omega), ..., \mathcal{W}_{\cdot n_t}^t(\omega)).$$

The MS-SLP formulation assumes that random observations at stage t, ξ^t, are independent of decisions made in prior stages. This assumption is key in establishing important properties discussed in section 2.3 that underpin solution methods. Since the only interaction between the stages is through the decision variables we can write the above problem more concisely as the following recursion:

$$Q^t(\mathbf{x}^{t-1}, \xi^t) = \min\{\mathbf{c}^t(\omega)\mathbf{x}^t(\omega^t) + Q^{t+1}(\mathbf{x}^t) \mid W^t(\omega)\mathbf{x}^t(\omega) = \mathbf{h}^t(\omega) - T^{t-1}(\omega)\mathbf{x}^{t-1},\ \mathbf{x}^t(\omega) \geq 0\} \tag{2.6}$$

where $Q^{t+1}(\mathbf{x}^t) = E_{\xi^{t+1}}[Q^{t+1}(\mathbf{x}^t, \xi^{t+1})]$ for $t = 1, ..., H$. The solution of (2.6) yields the optimal decision at each stage for each set of possible realizations at that stage.

2.2.3 Chance Constrained Programs

Another way of formulating a stochastic program is to define constraints that hold with a certain probability $\alpha \in [0, 1]$. In the above formulations of 2S-SLP and MS-SLP the second stage constraints are specified to hold with probability 1. A generalization is to specify *chance constraints* which hold with a specified probability. Such constraints are of the form

$$P\{F\mathbf{x} \geq \mathbf{g}(\omega)\} \geq \alpha \tag{2.7}$$

where $F\mathbf{x} \geq \mathbf{g}(\omega)$ is a set of linear constraints that must hold with probability at least α. This type of formulation was first discussed in the context of stochastic programming by Charnes and Cooper (1959). Chance constraints are often very useful for modeling risk. For example, in a manufacturing problem the probabilistic constraint might define the probability of a stock-out, in a portfolio optimization problem it might restrict a capital loss.

Models with constraints of the form in (2.7) are referred to as *p-models*. Other chance constrained formulations include the case in which the objective or constraints depend on the variance of some random variables (*V-model*) or a quantile of a random function. These

formulations are fundamentally different than the recourse models 2S-SLP and MS-SLP. However, in some special cases an equivalent 2S-SLP or MS-SLP model can be found for a given problem (see Gartska, 1980, for an example).

2.2.4 Nonanticipativity Constraints

There is an alternate formulation of 2S-SLP and MS-SLP based on redefining the decision variables and using *nonanticipativity constraints*. In this formulation a separate set of decision variables is defined at each stage for each scenario, and nonanticipativity constraints are used to enforce the fact that a decision made at a given stage should not anticipate the outcomes of random variables in later stages. For example, although for stage t the decision variable, $\mathbf{x}_t(\omega)$, is defined for every set of future realizations, it is required to satisfy the following additional constraint

$$\mathbf{x}_t(\omega) = E_{\Omega^t}[\mathbf{x}_t(\omega)].$$

where $\Omega^t = \{\omega^t \mid \omega_i \in \Omega_i,\ i = 1, ..., t\}$, i.e., $\mathbf{x}_t(\omega)$ must be independent of future random outcomes.

2.3 Properties of Stochastic Linear Programs

In this section some important properties of SLP's are summarized. The focus is on discussion of properties of 2S-SLP. Some extensions of the results for 2S-SLP to MS-SLP and important results for chance constrained problems are discussed at the end of the section.

2.3.1 Properties of 2S-SLP

From the structure of (2.4) it is clear that the first stage decisions, $\mathbf{x}$, affect the feasible region for second stage decisions. Thus there are two sets of constraints which define the feasible region for $\mathbf{x}$. The first set consists of deterministic first stage constraints, and are

denoted by

$$K_1 = \{x \mid A\mathbf{x} = \mathbf{b}\}.$$

The second is a set of *induced constraints* which result from the requirement that the second stage problem is feasible. There are different ways of specifying the induced constraints. Letting $Q(\mathbf{x}, \xi) = +\infty$ if the second stage problem is infeasible, the induced feasible region, K_2, is

$$K_2 = \{\mathbf{x} | Q(\mathbf{x}) < \infty\}.$$

An alternative definition, referred to as the possibility interpretation, is

$$K_2^p = \{\mathbf{x} \mid \forall \xi \in \Xi, \text{there exists } \mathbf{y} \geq 0 \ s.t. \ \mathcal{W}(\omega)\mathbf{y} = h(\omega) - \mathcal{T}(\omega)\mathbf{x}\}$$

Equivalence of K_2 and K_2^p is not guaranteed. For example, K_2 does not require feasible completion for all $\xi \in \Xi$; if Ξ is continuous, it may contain a countable set of points for which $Q(\mathbf{x}, \xi) = +\infty$ provided the set of points has probability zero. On the other hand it is possible that, for given $\mathbf{x}$, $Q(\mathbf{x}, \xi) < +\infty$, $\forall \xi \in \Xi$, but $E_\xi[Q(\mathbf{x}, \xi)]$ is unbounded. The following are fairly non-restrictive sufficient conditions for equivalence of K_2 and K_2^p, first derived by Wets (1974).

Proposition 1 : *The sets K_2 and K_2^p coincide if:*

i. Ξ is finite.

ii. Ξ is continuous, $\mathcal{W}$ is fixed (independent of ξ), and ξ has finite second moments.

See Wets (1974) for a proof.

Necessary conditions for equivalence are not known. However, in practice the above conditions are not very restrictive. Given these conditions the following propositions can be proved

Proposition 2 *Given either of the conditions in proposition 1 the following properties hold*

i. K_2 *is closed and convex.*

ii. if T *is fixed,* K_2 *is polyhedral.*

iii. if Ξ_T *represents the support of the distribution of* $T(\xi)$, $\mathrm{h}(\xi)$ *and* $T(\xi)$ *are independent, and* Ξ_T *is polyhedral, then* K_2 *is polyhedral.*

Proof of this proposition can be found in Wets (1974). Having defined some properties of the feasible region of $\mathbf{x}$ we now summarize some important properties of $Q(\mathbf{x})$.

Proposition 3 *If* $\mathcal{W}$ *is fixed then* $Q(\mathbf{x}, \xi)$ *is*

i. piecewise linear convex in $(\mathbf{h}(\xi), T(\xi))$.

ii. $Q(\mathbf{x}, \xi)$ *is piecewise linear concave in* $\mathbf{q}$.

iii. $Q(\mathbf{x}, \xi)$ *is piecewise linear convex in* $\mathbf{x}$ *for all* $\mathbf{x} \in \{K_1 \cap K_2\}$.

Problems in which $\mathcal{W}$ is fixed are referred to as *fixed-recourse* problems. The properties in propositions 1-3 follow from the fact that $Q(\mathbf{x}, \xi)$ is a linear program. (Proofs can be found in Birge and Louveaux (1997), Chapter 3, Theorem 6.) In some cases the feasible regions K_1 and K_2 may have special properties which are useful in computation. For example, if $K_1 \subset K_2$, then every first stage decision is second stage feasible. Problems which exhibit this structure are said to have *relatively complete recourse.* It is often difficult to verify this property; however, many formulations satisfy a weaker condition, referred to as *complete recourse.* This refers to the condition that the recourse matrix, $\mathcal{W}$, contains a positive linear basis, i.e., $pos(\mathcal{W}) = \Re_2^m$.

Properties in propositions 1-3 can be used to prove the following properties of the expectational functional $Q(\mathbf{x})$.

Proposition 4 *If* $\mathcal{W}$ *is fixed and* ξ *has finite second moments*

i. $Q(\mathbf{x})$ *is Lipschitzian convex and finite on* K_2.

ii. When Ξ *is finite* $Q(\mathbf{x})$ *is piecewise linear.*

iii. If $F(\xi)$ *is a continuous distribution then* $Q(\mathbf{x})$ *is differentiable on* K_2.

Proof: convexity and finiteness of $Q(\mathbf{x})$ follow directly from the conditions in propositions 1 and 3, and piecewise linearity when Ξ is finite also follows from proposition 3. The definition of a Lipschitzian convex function, and proofs of the continuity properties appear in Wets (1974; theorem 7.7 and proposition 7.18 respectively).

An alternate representation of $Q(\mathbf{x})$ that uses the dual of $Q(\mathbf{x}, \xi)$ is

$$Q(\mathbf{x}) = \int_{\Xi} \pi(\mathbf{x}, \xi)(\mathbf{h}(\xi) - \mathcal{T}(\xi)\mathbf{x}) dP(\xi). \tag{2.8}$$

where $\pi(\mathbf{x}, \xi)$ denotes the optimal solution to the dual of $Q(\mathbf{x}, \xi)$. The following proposition is important in the context of decomposition algorithms.

Proposition 5 *Given the conditions in proposition 1 the hyperplane*

$$\{(z, \mathbf{x}) \mid z + E_{\xi}[\pi(\bar{\mathbf{x}}, \xi)\mathcal{T}(\xi)]\mathbf{x} = E_{\xi}[\pi(\bar{\mathbf{x}}, \xi)\mathbf{h}(\xi)]\}$$

is a supporting hyperplane of $Q(\mathbf{x})$ *at* $\bar{\mathbf{x}}$.

A proof of proposition 5 appears also in Wets (1974). Proposition 5 plays an important role in solution algorithms such as the *L-shaped method* (Van-Slyke and Wets, 1969) discussed in section 2.4.

Conditions under which stochastic linear programs are dualizable can be found in Rockafellar and Wets (1976). Also, conditions under which solutions are attainable, and issues regarding stability, are reviewed in Chapter 3 of Birge and Louveaux (1997).

2.3.2 Properties of MS-SLP

Many of the properties of the 2S-SLP can be extended to MS-SLP due to its recursive structure. Given the sequential structure of the problem, the stage-t feasible region can be defined by

$$K_t = \{\mathbf{x}^t \mid Q^{t+1}(\mathbf{x}^t) < \infty\}.$$

Similar properties for the equivalence of K_t to a possibility interpretation of the feasible region can be derived. From a practical standpoint most multi-stage models assume finite Ξ. The following is an important result for the development of solution methods for MS-SLP.

Proposition 6 *The sets K_t and stage-t recourse functions $Q^{t+1}(\mathbf{x}^t)$ are convex $\forall$ t.*

Proof: see Birge and Louveauz (1997).

See Birge (1985), Gassman (1990), and Birge and Louveaux (1997), Chapter 11, for additional details of MS-SLP.

2.3.3 Probabilistic Constraints

Probabilistic constraints create additional difficulties not present in 2S-SLP or MS-SLP. For instance, the constraint regions for problems formulated with chance constraints may be nonconvex and even disconnected. A class of measures for which chance constraints lead to well defined SLP's is the class of *quasi-concave* measures. A measure P is *quasi-concave* if, for every convex measurable sets U and V it satisfies

$$P((1-\lambda)U + \lambda V) \geq \min\{P(U), P(V)\}$$

where $0 \leq \lambda \leq 1$. Letting $K(\alpha) = \{\mathbf{x} \mid P(A\mathbf{x} \geq \mathbf{g}) \geq \alpha\}$ denote the set of feasible $\mathbf{x}$, the above property can be summarized as

Proposition 7 *If $\mathbf{g}$ has quasi-concave measure, then $K(\alpha)$ is a closed convex set for $0 \leq \alpha \leq 1$.*

A proof of proposition 7 and additional properties of chance constraints appear in Prékopa (1980). Quasi-concave measures are a fairly general class which includes distribution functions such as uniform, normal, and lognormal.

2.4 Exact Methods for Stochastic Linear Programs

In this section we discuss solution methods that can be applied when the support, Ξ, is finite. In this case 2S-SLP is an LP with block-diagonal structure. We briefly review some of the solution methods that take advantage of this structure: extreme point, decomposition, quasi-gradient, interior point, and general methods based on nonlinear optimization.

2.4.1 Extreme Point Methods

When Ξ is finite, stochastic linear programs can be solved using the simplex method. However, extreme point methods such as this require factorization of the basis matrix, B, and as the number of possible realizations, K, increases, factorization by standard methods eventually becomes impractical. (Note that K is the cardinality of Ξ, which we also denote as $|\Xi|$.) The block-angular structure of the constraint matrix can be exploited to achieve computational savings. For example, consider the following proposition (adapted from Birge and Louveaux (1997)).

Proposition 8 *By permuting rows of the basis matrix, B, of a 2S-SLP it can be put in the following form*

$$B' = \begin{bmatrix} D & C \\ F & L \end{bmatrix}$$

where D is a square invertible matrix of order n_1 and L is a block-diagonal matrix with K invertible blocks, each with order at most m_2. Thus we have the following equivalent system of equations

$$D\mathbf{x}_B + C\mathbf{y}_B = \mathbf{b}', \; F\mathbf{x}_B + L\mathbf{y}_B = \mathbf{h}', \tag{2.9}$$

where $\mathbf{b}' = \begin{bmatrix} \mathbf{b} \\ \mathbf{h}_u \end{bmatrix}$, $\mathbf{h}' = \mathbf{h}_l$, and $\mathbf{h}_u$ represents the right hand side vectors for rows of $\mathcal{T}$ in D, and $\mathbf{h}_l$ represents the remaining components with rows in F. Since L is invertible we can use (2.9) to write

$$\mathbf{y}_B = L^{-1}(\mathbf{h}' - F\mathbf{x}_B) \tag{2.10}$$

and using (2.9) we get

$$(D - CL^{-1}F)\mathbf{x}_B = \mathbf{b}' - CL^{-1}\mathbf{h}'. \tag{2.11}$$

Computational saving results from the fact that L is block-diagonal. Extreme point methods that take advantage of this were developed for the two-stage problem (Kall, 1979, and Strazicky, 1980) and also for the multi-stage problem (Birge, 1980).

2.4.2 Interior Point Methods

Interior point methods also benefit from the block-diagonal structure of the constraint matrix in 2S-SLP. For example, consider a simplified version of Karmarkar's (1984) algorithm for solving

$$\min\{\mathbf{cx} \mid \bar{A}\mathbf{x} = \mathbf{b}\}.$$

The method is based on iteratively moving along descent directions which are computed by projecting the gradient $\mathbf{c}$ onto the feasible region $\{\mathbf{x} \mid \bar{A}\mathbf{x} = \mathbf{b}\}$. This is done using the following *projection matrix*

$$P = I - \hat{A}^T(\hat{A}\hat{A}^T)^{-1}\hat{A} \tag{2.12}$$

where $\hat{A} = D\bar{A}$ and D is a matrix with elements $\{x_1^\nu, ..., x_n^\nu\}$ along the diagonal and zero otherwise. Typically most of the computational work done at an iteration is in computing $(\hat{A}\hat{A}^T)^{-1}$. Although this matrix is dense a factorization scheme which takes advantage of this structure was developed by Birge and Qi (1988). Birge (1985) also discusses the use of interior point methods for the solution of multi-stage problems and its relations to

decomposition based approaches. Carpenter, Lustig and Mulvey (1991) show how various formulations of 2S-SLP, using nonanticipativity constraints, can be used to yield a matrix M with structure that can be exploited to achieve efficient factorization. Choi and Goldfarb (1993) demonstrate how a primal-dual path-following algorithm can exploit the structure of a block-diagonal constraint matrix, and Bahn et. al. (1995) investigate the use of analytic center cutting-plane methods for solving 2S-SLP.

2.4.3 Decomposition Methods

Basic decomposition approaches include Dantzig-Wolfe decomposition (Dantzig and Wolfe, 1960) and Benders decomposition (Benders, 1962), also referred to as dual and primal decomposition, respectively. It is the latter which formed the basis for the well known L-shaped algorithm (Van Slyke and Wets, 1969). Many of the existing algorithms for stochastic linear programming are adaptations and extensions of the L-shaped method.

L-Shaped Algorithm for 2S-SLP

The basic idea of the L-Shaped algorithm is to decompose the problem into (a) a master problem in the first stage decision variables, and (b) subproblems in second stage decision variables. Subgradient information from the subproblems is used to generate supporting hyperplanes (optimality cuts) to outer linearize $Q(\mathbf{x})$. The algorithm is based on solution of the following equivalent problem

$$\min\left\{\mathbf{c}^T\mathbf{x}+\theta \mid A\mathbf{x}=\mathbf{b}, Q(\mathbf{x}) \leq \theta,\ \mathbf{x} \geq 0\right\}. \tag{2.13}$$

The basic form of the algorithm is as follows:

step 0 : set the number of iterations, v, the number of feasibility cuts, r, and the number of optimality cuts, s, to zero.

step 1 : solve the master problem

$$\min\{\mathbf{cx}+\theta\}$$

s.t.

$$A\mathbf{x} = \mathbf{b} \tag{2.14}$$

$$D_l\mathbf{x} \geq \mathbf{d}_l \quad l = 1, ...r \tag{2.15}$$

$$E_l\mathbf{x} + \theta \geq \mathbf{e}_l \quad l = 1, ...s \tag{2.16}$$

$$\mathbf{x} \geq 0, \quad \theta \in \Re.$$

Let $(\mathbf{x}^v, \theta^v)$ be solution. The constraint set (2.14) is the set of first stage constraints, (2.15) is the set of feasibility cuts and (2.16) is the set of optimality cuts.

step 2 : solve the following *phase-I* LP for $k = 1, ..., K$

$$w = \min\{\mathbf{ev}^+ + \mathbf{ev}^- \mid \mathcal{W}\mathbf{y} + I\mathbf{v}^+ - I\mathbf{v}^- = \mathbf{h}_k - \mathcal{T}_k\mathbf{x}^v,\ \mathbf{y}, \mathbf{v}^+, \mathbf{v}^- \geq 0\} \tag{2.17}$$

where $\mathbf{e} = (1, ..., 1)$, until k is found such that $w > 0$. Add the following feasibility cut to the master problem

$$D_{r+1} = \sigma^v\mathcal{T}_k,\ \mathbf{d}_{r+1} = \sigma^v\mathbf{h}_k.$$

where σ^v is the dual solution of the *phase-I* LP. Add $D_{r+1}\mathbf{x} \geq \mathbf{d}_{r+1}$ to the master, set $r = r + 1$, and return to *step 1*. If $w = 0, \forall k$ then go to *step 3*.

step 3 : solve subproblems for $k = 1$, ... K and generate the optimality cut defined by

$$E_v = \sum_{k=1}^{K} p_k(\pi_k^v)^T\mathcal{T}_k,\ e_v = \sum_{k=1}^{K} p_k(\pi_k^v)^T\mathbf{h_k}.$$

Set $s = s + 1$.

step 4 : If $E_v\mathbf{x}^v + \theta \geq \mathbf{e}_v$ is satisfied then the current solution is optimal. Otherwise add the new optimality cut to the master problem and return to step 1.

It can be proved that the algorithm converges to the optimal solution in a finite number of steps (Van Slyke and Wets, 1969).

Methods have been proposed for increasing the efficiency of the L-shaped algorithm. Birge and Louveaux (1988) suggested a multi-cut version of the algorithm in which a cut is added to the master for each subproblem rather than a single aggregate cut. A method

referred to as *bunching* was discussed by Wets (1988) and Gassman (1990): it reduces computation time in *step 3* by bunching subproblems that have the same optimal basis. A method which utilizes the framework of the L-shaped method and the recursive structure of MS-SLP to solve multi-stage problems is the *nested decomposition method* based on the algorithm of Ho and Manne (1974) for deterministic models. Cuts on the recourse function are added at each stage, $Q^{t+1}(\mathbf{x}^t)$, that result in feasible completion in all future stages, as well as improved lower bounds. The main difference is that now there is a master problem for each stage, and each scenario at that stage. At each iteration the algorithm requires that several stages be traversed; thus the computational effort increases greatly with the number of stages as well as the number of scenarios. Several different criteria for moving among the time stages have been suggested and results of numerical experiments that test a number of these were reported by Gassman (1990).

2.4.4 Lagrangian Based Approaches

All of the methods discussed so far are based on linear programming methods. Lagrangian based methods, on the other hand, are based on nonlinear optimization techniques. They typically utilize the nonanticipativity constrained formulation of stochastic linear programs. The fact that the nonanticipativity constraints are the only linking constraints in the problem is the basis behind these approaches. To illustrate consider the 2S-SLP with Ξ consisting of K scenarios, each with probability p_k. The dual program is

$$\max_{\pi}\{\theta = \min\{\sum_{k=1}^{K} p_k[\mathbf{cx} + Q(\mathbf{x}_k, \mathbf{y}_k) + \pi_k(\mathbf{x}_k - \sum_{k=1}^{K} p_k\mathbf{x}_k)]\}\}. \tag{2.18}$$

A basic gradient method for (2.18) is

step 0. Set π^0, $\nu = 0$ and go to step 1.

step 1. Let $\pi = \pi^\nu$ and solve (2.18) for $(x_1^\nu, ..., x_K^\nu, y_1^\nu, ..., y_K^\nu)$.

step 2. If $\mathbf{x}_k = \sum_{k=1}^{K} p_k\mathbf{x}_k$, $\forall k$ then stop with an optimal solution. Otherwise set $\hat{\pi}_k =$

$\mathbf{x}_k - \sum_{k=1}^{N} p_k \mathbf{x}_k$ and go to step 3.

step 3. Determine λ^ν that minimizes $\theta(\pi^\nu + \lambda\hat{\pi})$ such that $\pi^\nu + \lambda\hat{\pi} \geq 0$, $\lambda \geq 0$. Let $\pi^{\nu+1} = \pi^\nu + \lambda^\nu\hat{\pi}$ and go to step 1.

For finite Ξ the algorithm converges to the optimum in a finite number of iterations. Computational saving results from the reduced computational burden in step 1 because the nonanticipativity constraints appear in the objective. If the number of iterations is small then improvements over methods that evaluate $Q(\mathbf{x}_k, \mathbf{y}_k)$ directly can be expected. Another Lagrangian based approach, called *progressive hedging*, and developed by Rockafellar and Wets (1991), achieves complete separation of the scenario subproblems. This results in substantial computational saving at each iteration but may result in an increased number of iterations. Computational advantages from using this algorithm are reported by Mulvey and Vladimmirou (1991) for stochastic network problems. Lagrangian based approaches have found recent application to solution of 2S-SLPs and MS-SLPs with discrete decision variables (e.g. Takriti and Birge, 2000, Caroe et al., 1997, Takriti and Birge, 1995).

2.5 Approximations

The key difficulty that every stochastic programming problem faces is the evaluation of the expectation function $Q(\mathbf{x})$. Previous sections discussed methods applicable to the case of finite Ξ, most of which were large scale linear programming methods that exploit the common problem structure of stochastic linear programs. However, as the number of scenarios becomes large or for the case of continuous support, Ξ, approximations are necessary. Not surprisingly most approximations are based on using a limited number of scenarios. We briefly mention some of the approaches discussed in the literature.

2.5.1 Quasi-Gradient Methods

The *stochastic quasi-gradient* methods use sampling to approximate solutions to SLPs. They are based on the application of nonlinear optimization. A sequence of approximates, $\mathbf{x}^v, v = 0, 1, ...,$ is generated by solving the approximate optimization problem, called the *sample average approximation*, defined by statistical estimates of $Q(\mathbf{x}^v)$ and its gradient, $Q_x(\mathbf{x}^v)$:

$$\bar{Q}(\mathbf{x}^v) = \frac{1}{N_v}\sum_{k=1}^{N_v} Q(\mathbf{x}^v),\ Q_x(\mathbf{x}^v) = \frac{1}{N_v}\sum_{k=1}^{N_v} Q_x(\mathbf{x}^v)$$

where N_v is the sample size. Sampling may be carried out using crude monte-carlo sampling or using more sophisticated methods such as importance sampling. Problems are solved iteratively and at each iteration the sample size is increased. Under assumptions that are nonrestrictive from a practical point of view, these methods can be proved to converge with probability 1. However, difficulties arise in their practical application because finite sample sets are used. Stopping rules are often statistical in nature and based on estimates of confidence intervals around the true optimal solution. Several different methods have been studied and a detailed review can be found in Ermoliev (1988).

2.5.2 Decomposition Based Sampling Methods

A disadvantage of quasi-gradient methods is that computational effort may be wasted as a result of inaccurate approximation because an approximate optimum is found at each iteration. Improved results may be obtained by using sampling within decomposition based algorithms such as the L-shaped algorithm where complete optimization is not carried out at each step. Dantzig and Glynn (1990) were among the first to use such an approach. At each iteration of the algorithm, $Q(\mathbf{x}, \xi)$ is sampled and approximate optimality cuts are generated. For each sample j,

$$Q(\mathbf{x}, \xi^j) \geq Q(\mathbf{x}^\nu, \xi^j) + \pi^j(\mathbf{x}^\nu)(\mathbf{x} - \mathbf{x}^\nu),$$

by convexity of $Q(\mathbf{x}, \xi)$ and the subgradient inequality, and therefore

$$\frac{1}{\nu}\sum_{j=1}^{\nu} Q(\mathbf{x}, \xi^j) \geq \frac{1}{\nu}\sum_{j=1}^{\nu} Q(\mathbf{x}^\nu, \xi^j) + \frac{1}{\nu}\sum_{j=1}^{\nu} \pi^j(\mathbf{x}^\nu)(\mathbf{x} - \mathbf{x}^\nu). \tag{2.19}$$

By the central limit theorem

$$\frac{1}{\nu}\sum_{j=1}^{\nu} Q(\mathbf{x}^\nu, \xi^j) + \frac{1}{\nu}\sum_{j=1}^{\nu} \pi_k^j(\mathbf{x} - \mathbf{x}^\nu)$$

is asymptotically normally distributed, with mean

$$Q(\mathbf{x}^\nu) + E_\xi[\pi^j(\mathbf{x}^\nu)](\mathbf{x} - \mathbf{x}^\nu)$$

and (2.19) converges to a valid optimality cut at $\mathbf{x}^\nu$ as $\nu \to \infty$. However, in practice a finite sample size is used and the above cut is an approximation. A cut generated using sampling can be thought of as including some normally distributed error term, $\epsilon_\nu(\mathbf{x})$, as follows

$$Q(\mathbf{x}) \geq \sum_{j=1}^{\nu}(Q(\mathbf{x}^\nu, \xi^j) + \pi^j(\mathbf{x}^\nu)(\mathbf{x} - \mathbf{x}^j)) + \epsilon_\nu(\mathbf{x}). \tag{2.20}$$

Feasibility cuts are generated whenever a sample results in infeasibility in the second stage. Stopping criteria, based on confidence intervals, rely on certain assumptions about the cuts generated. Infanger (1992) showed that significantly tighter confidence intervals can be obtained using importance sampling, and the assumption that consecutive optimality cuts are independent. The concern when using this approach is that the error term in (2.20) may lead to inaccurate outer linearization of the recourse function and subsequent inaccuracy in the feasible minimum solution found.

A method which combines the concepts of quasi-gradient methods and decomposition, called stochastic decomposition, was developed by Higle and Sen (1991) for 2S-SLP. Cuts are generated using small sample sizes. Thus initially there may be significant inaccuracy in the outer linearization of the recourse function. However, the cuts are adjusted at each iteration such that they drop away as the algorithm proceeds. It can be shown that the algorithm generates a sequence of iterates that converges to the optimal solution with

probability 1. Shapiro and Homem-deMello (1998) review other sampling based methods that utilize partial (non-linear) optimization in iterative algorithms in which sample size is progressively increased.

2.5.3 Deterministic Bounds

Methods that are based on deterministic bounds can be used to compute bounds on $Q(\mathbf{x})$ which can subsequently be used in an algorithm to approximate the solution to an SLP. Typical methods replace the true measure P with a discrete measure for which $Q(\mathbf{x})$ can be computed efficiently. For example, one common way of discretizing P is to use a partition $S^{(\nu)} = \{S^k,\ k = 1, ..., \nu\}$ where

$$S^k = [a_1^{(k)}, b_1^{(k)}] \times [a_2^{(k)}, b_2^{(k)}] \times \cdots \times [a_n^{(k)}, b_n^{(k)}]$$

is a rectangular partition, and the discrete distribution is defined by the conditional probabilities, p_k, and conditional expectations, $\boldsymbol{\xi}^k$, over the cells of the partition. From this the Jensen inequality can be used to obtain the following lower bound

$$E[Q(\mathbf{x}, \boldsymbol{\xi})] \geq \sum_{k=1}^{\nu} p^k Q(\mathbf{x}, \boldsymbol{\xi}^k).$$

Methods for computing upper bounds on the recourse function also exist. They are often based on finding an extremal measure such that the resulting expectation is an upper bound. Thus such methods require that Ξ be compact for the bound to be defined. An example of a classic upper bound is the Edmundson-Madansky (EM) upper bound and it's generalization over a partition (Huang, Ziemba and Ben-Tal, 1977). It generates an upper bound as a weighted sum of the function at extreme points of Ξ. For example, in one dimension, if $\Xi = [a, b]$, the bound is

$$(\frac{b - \bar{\xi}}{b - a})Q(\mathbf{x}, a) + (\frac{\bar{\xi} - a}{b - a})Q(\mathbf{x}, b).$$

The above bound can be easily extended to multiple dimensions when the random variables are stochastically independent. Frauendorfer (1988b) provided the generalization of the EM bound to the dependent case. The use of Jensen and EM bounds in an adaptation of the L-shaped algorithm is discussed by Frauendofer and Kall (1988). As the number of random variables in a problem grows, the number of extreme points of Ξ increases exponentially. Thus computation of EM bounds quickly becomes intractable. Fraundorfer (1992) discusses a tractable bound in high dimensions based on using *simplices* that contain Ξ. Also, a method based on solution of the generalized moment problem was developed by Birge and Wets (1987).

2.6 Applications

The first application of stochastic programming was to airline fleet assignment, by Ferguson and Dantzig (1956). Since then stochastic programming models have been applied to problems in areas including production planning, facility location, scheduling power systems, capacity planning, transportation systems, and project scheduling. The above list is not exhaustive; a more comprehensive discussion of stochastic programming applications can be found in King (1988). The following are more recent examples.

Due to their long term nature, capacity planning decisions require significant analysis of potential future outcomes and possible recourse actions. Such decisions are often *irreversible*, in the sense that once they are made the new capacity can not simply be re-sold if it is found to be underutilized. Eppen et al. (1988) discuss a capacity planning model for making automotive assembly capacity investment decisions at General Motors. The goal was to determine capacity investments, for a range of different products, that maximize expected profit subject to a downside risk constraint. The model captures uncertainty in future demand, and various recourse actions such as temporary or permanent shutdown,

and/or reconfiguration or retooling of facilities.

There are many articles in the literature on applications to the optimization of power systems. For instance, Pereira and Pinto (1991) develop solution methods for an MS-SLP model of the Brazillian power system. They model the control of reservoirs, given uncertainty in loads on the system, and uncertainty in reservoir levels due to rainfall. Takriti, et al. (1995) present a model for daily scheduling of the Michigan State power system which includes binary decision variables for modeling the decision whether to switch a particular unit on or off. Caroe et al. (1997) consider a similar type of model for electrical load planning, given multiple types of power generation units (conventional coal, gas fired thermal units, and hydro-electric plants).

An important application to economic policy is the management of environmental projects. Somlyody and Wets (1988) discuss models for analyzing policies designed to reverse the deterioration of water quality in lakes. Pinter (1991) discusses a broad range of models for identifying, estimating, and controlling the effects of various processes (both natural and artificial) on environmental systems.

More recently there has been increasing interest in the application of stochastic programming models to finance. For example, Carino and Ziemba (1998) and Carino et al. (1998) describe the application of an MS-SLP model for determining the appropriate asset and liability mix over time for a Japanese insurance company. The objective there is to maximize expected profit, given uncertain returns on investments and various constraints on the asset and liability mix over time resulting from cash flow requirements, legal obligations, and taxation.

2.7 Future Research Directions

Several exact algorithms that take advantage of the structure common to stochastic linear programs are reviewed above. These can be used to find exact solutions to general, and often large-scale, problems. However, many real-world problems are well beyond the current reach of exact methods. Thus the study and development of approximations is a necessity, both to broaden the range of problems that can be suitably modeled, and to reduce computation times to make *real-time* applications feasible. To achieve improvements in existing methods based on deterministic bounds it will be necessary to exploit specific problem structures. On the other hand, statistical sampling based methods hold the hope of efficient algorithms for generating high confidence solutions.

An area of growing interest is the study of solution methods for stochastic linear programs, with discrete decision variables (see van der Vlerk, 2000, for a current bibliography, and Schultz et al., 1996, for a review). Many realistic problems require discrete decision variables to model 0-1 decisions describing whether or not to take a certain action. Given the computational burden of the typically large size of stochastic programs the inclusion of such decision variables puts extreme constraints on the implementation of solution methods. Thus there is a need for efficient heuristics that take advantage of problem structures to achieve feasible and near optimal solutions in reasonable computation times.

Chapter 3

Appointment Scheduling Systems

3.1 Introduction

Appointment systems are used in many customer service industries to increase the utilization of resources, match workload to available capacity, and smooth the flow of customers. A common problem faced by decision makers is how to determine the scheduled start times of services when their durations are uncertain. This problem is materially different from say a machine scheduling problem (for example, see Forst, 1993) in the sense that once appointments are set, customers are not available prior to the scheduled start time, even if the server becomes free at an earlier time. Thus, choosing an early start time will lead to better server utilization at the cost of additional waiting by customers, whereas a late start time will reduce customer waiting at the cost of additional server idling. What we propose in this Chapter is a model that can be used to find optimal start times under different cost structures associated with server idling, customer waiting, and tardiness.

The appointment scheduling problem arises in many contexts and appointment decisions are economically significant. For example, Sabria and Daganzo (1989) consider it from the viewpoint of scheduling the arrival of cargo ships at a seaport. In their treatment of the problem the costs of underutilization of a seaport are traded off against the cost of cargo ship

waiting. Wang (1994) discusses the problem in a manufacturing setting where the objective is to schedule the arrival of parts on the shop floor such that work-in-process inventory and server idling are minimized. Also, there have been numerous studies presented in operations research, statistics, and health care journals over the past three decades on the problem of assigning appointments for arrivals at outpatient clinics (for example Bailey, 1952, Welch, 1964, Jansson, 1966, Soriano, 1966, Mercer, 1973, Charnetski, 1984, Ho and Lau, 1992, Dexter, 1999 and references therein). We begin by motivating the problem in another light by providing a specific example in the context of allocating resources for elective surgeries at hospitals.

Typical urban hospitals in North America have annual operating expenditures measured in hundreds of millions of dollars. Operating rooms (ORs) are estimated to account for between a third and a half of the total costs incurred by hospitals (Redelmeier and Fuchs, 1993, and Macario et al., 1995). As a result, ORs represent an area with high potential for cost savings. Even small relative improvements in efficiency translate into significant dollar savings and benefits to society. Major components of OR costs are fixed costs. These consist of salaries of staff (surgeons, anesthesiologists, nurses, and technicians) and a fixed cost of facilities and equipment. Thus, effective delivery of surgical services requires an OR manager, or similar governing body, to schedule surgeries efficiently so as to tradeoff high utilization of the OR staff and other resources with low OR idling and overtime costs. In most large hospitals in North America, a *block-booking* strategy is used. This strategy involves allocating a contiguous block of time for a sequence of surgeries within a department (typically performed by the same surgeon) at a particular OR. Based on the allocation of block times, a particular time of day is specified for the arrival of the staff and material resources to the OR. Since surgery durations are not known with certainty it is common practice to schedule block sizes based on estimates of the sum of mean durations of surgeries in the block. Typically overtime costs are avoided by scheduling an empty block at the end

of the day to absorb fluctuations in finish times of all blocks scheduled that day. Using a numerical example, we show in this Chapter that an optimal block schedule can effectively increase the capacity of an OR and at the same time lower the sum of expected waiting, overtime and idling costs. Thus, the optimization model discussed in this Chapter can be the source of significant cost savings. Other issues surrounding the scheduling of elective surgeries at hospitals are discussed in detail in papers by Goldman, Knappenberger and Shearson (1970), Pierskalla and Brailer (1994), Strum, Vargas and May (1999) and Dexter et al. (1999).

We shall use the term *customers* to refer to resources (e.g., surgical teams in the block-booking example above) which become available only at the assigned start time, and *facility* or *server* to refer to fixed resources, such as an OR. The evaluation of a given schedule of appointment times requires the calculation of expected customer waiting times and facility idle times. Exact calculation of these quantities is problematic when there are many jobs because it requires the evaluation of multidimensional integrals. Many previous studies have used simulation to study the performance of heuristic rules for setting appointments. According to one heuristic regime (Bailey, 1952 and 1954, Welch and Bailey, 1962, and Welch, 1964), if there are n customers to be scheduled, m of them are scheduled to arrive at the beginning of the session and the remaining $n - m$ appointment times are spaced by their mean job durations (denoted by μ_is). Thus, if a_i represents the appointment time for job i then

$$a_i = 0, \qquad i = 1, \ldots, m, \tag{3.1}$$

$$a_i = a_{i-1} + \mu_i, \quad i = m+1, \ldots, n. \tag{3.2}$$

Alternatively, in the block appointments regime (White and Pike, 1964, Soriano, 1966), a session of length d is broken up into k blocks and $n_j = n/k$ customers are scheduled to arrive at the start of each block. Thus if we let index i denote the customer, and j the

block to which the customer belongs, then

$$a_{ij} = jd/k, \quad i = 1, \ldots, n, \text{ and } j = 1, \ldots, k. \tag{3.3}$$

Heuristics for assigning individual appointment times to customers have also been explored. For example, Charnetski (1984) considered a heuristic that assigns a job allowance of $\mu_i + h\sigma_i$ to job i, regardless of its place in the sequence where μ_i and σ_i denote the mean and standard deviation of the i^{th} job duration, respectively. He experimented with different values of h using a simulation model while assuming that job durations are normally distributed. Ho and Lau (1992) also used simulation to compare the performances of a number of heuristics. Robinson and Chen (2000) study empirical data based heuristics developed from solutions obtained using a simulation based conjugate gradient method.

Articles by Jansson (1966), Mercer (1960 and 1973), Sabria and Daganzo (1989), and Brahimi and Worthington (1991) use queuing analysis to study the same problem. This literature generally assumes that job durations are independent and identically distributed (i.i.d.), and appointment times are equally spaced. With respect to the latter assumption, it has been shown that the optimal spacing of appointments (job allowances) when service times are i.i.d. is not in general uniform (Wang 1993). Also, a majority of queuing theoretic models obtain expected customer waiting times and expected facility idle times under the assumption of a steady state. Often, however, appointments need to be set for finite session length d during which the facility is up and running, and a steady state is never reached in such cases.

Another line of research is the study of optimization models for appointment systems. Weiss (1990), and Robinson, Gerchak, and Gupta (1996), deal with two and three customer problems, respectively, which can be solved relatively easily owing to the low dimensionality of the problem. Robinson et al. also report a method based on Monte Carlo simulation for computing appointment times when $n > 3$. Wang (1993) considered the case in which job

durations are exponentially distributed and showed that for this special case the probability density function (p.d.f.) for customer waiting times is *phase-type*. He then exploited the computational advantages associated with phase-type distributions to find the optimal appointment times. Through numerical examples, he showed that optimal job allowances have a dome shape, i.e., they are initially increasing and then decreasing.

In this Chapter we formulate the appointment scheduling problem (ASP) as a two-stage stochastic linear program (2-SLP). It immediately follows that under nonrestrictive conditions, the ASP is a convex minimization problem (Birge and Louveaux, 1997). (Note that no previous study on appointments scheduling has actually managed to prove that ASP is convex for an arbitrary number of jobs and job durations.) Next, we develop an algorithm that utilizes the problem structure to obtain a near-optimal approximate solution. Our algorithm also obtains upper and lower bounds on the difference between the optimal and approximate solution. It has the property that the solution can be made arbitrarily close to the optimum by increasing the number of iterations performed. In applications where job sequence is not pre-ordained, it is of interest to determine an optimal job sequence, a task that is complicated by its combinatorial nature. We provide an upper bound on the expected cost savings that can be realized from resequencing and identify parameter values for which these savings are large/small.

Since variability in job durations is a major source of inefficiency in the use of OR time, we also study the effect of changing variance of job-durations, while keeping their means fixed, on the total expected cost of running an appointments-based service system. We show that expected costs increase linearly with standard deviation of job durations.

Even with our algorithm, solving problems involving large number of jobs can be time consuming. Furthermore, in OR and outpatient scheduling problems, a particular day's schedule may have to be revised several times based on cancellations and the arrival of more urgent patients. We therefore study several approximations and easy-to-implement

heuristics. Through extensive numerical experimentation, we report on the accuracy of these heuristics and on how optimal job allowances depend on parameters like skewness of job duration distributions, ratio of unit waiting to unit overtime costs, and the number of jobs scheduled per day.

The Chapter is organized as follows. The next section discusses different performance criterion for an appointment scheduling system, and introduces the SLP model for determining individual appointment times. section 3.3 presents the algorithm and bounds on its performance. section 3.4 discusses approximations for large problems and two heuristics that follow from them, and section 3.5 discusses insights and presents numerical examples to illustrate the practical importance of the model. Finally, section 3.6 summarizes implications for policy makers and discuss future research directions.

3.2 Formulation and Preliminary Analysis

We consider a single server system at which customers arrive punctually at scheduled appointment times, and are served in the order of their arrival. Job sequence is thus assumed fixed. We use the following notation throughout the Chapter:

n:	number of jobs to be scheduled.
$\mathbf{x}$:	vector of job allowances for the first $n-1$ jobs.
$\mathbf{a}$:	vector of scheduled start times for n jobs.
$\mathbf{Z}$:	vector of random job durations.
μ:	vector of mean durations for the n jobs.
$\mathbf{W}$:	vector of customer waiting times for given $(\mathbf{x}, \mathbf{Z})$.
$\mathbf{S}$:	vector of facility/server idle times between consecutive jobs for given $(\mathbf{x}, \mathbf{Z})$, e.g. S_2 is the idle time between jobs 1 and 2.
d:	time allotted for a given sequence of jobs (e.g. length of day or surgeon

block time for OR).

L: tardiness for a given sequence of jobs with respect to d for given $(\mathbf{x}, \mathbf{Z})$.

G: earliness for a given sequence of jobs with respect to d for given $(\mathbf{x}, \mathbf{Z})$.

$\mathbf{c}^w$: vector of cost coefficients for customer waiting.

$\mathbf{c}^s$: vector of cost coefficients for facility idle time.

c_ℓ: cost coefficient for tardiness with respect to d.

Bold face and upper case notation indicate vectors and random variables, respectively (to avoid confusion with lower case L we use script ℓ). The vector of job allowances $\mathbf{x} \in \Re^{n-1}$ (we need to specify only the job allowances for the first $n-1$ jobs), the vectors $\mathbf{a}, \boldsymbol{\mu}, \mathbf{Z}, \mathbf{W}, \mathbf{S}, \mathbf{c}^w, \mathbf{c}^s \in \Re^n$, and d, L, G, and c_ℓ are scalar quantities. The vector of random job durations, $\mathbf{Z}$, has support $\Xi \subseteq \Re^n$ and probability distribution P on $\Re^n$ and it is assumed that $\mathbf{Z}$ has finite first moments. The scheduled start time for a given job is equal to the sum of the job allowances of its predecessors. We assume that the first job commences at time zero, i.e., $a_1 = 0$ and $a_i = \sum_{j=1}^{i-1} x_j$ for $i = 2, ..., n$. The vectors of cost coefficients, $\mathbf{c}^w$ and $\mathbf{c}^s$, can be different for each job. For example, if there are different customer classes this can be modeled by having customer dependent waiting time costs.

Three commonly used metrics for the performance of an appointment system are customer waiting time, server idle time, and tardiness of a collection of jobs with respect to the alloted time for the session. Whereas early arrival increases customer waiting, late arrival results in increased idle time of the facility and greater overtime costs. A manager of an appointments-based service system needs to balance efficient server utilization against the cost of customer waiting and overtime. The relative weights of the different metrics may vary from one system to another. For a given realization of job durations, $\mathbf{Z}$, and job

allowances, x, these metrics can be written as the following recursions:

$$W_i = (W_{i-1} + Z_{i-1} - x_{i-1})^+, \quad i = 2, \ldots, n. \tag{3.4}$$

$$S_i = (-W_{i-1} - Z_{i-1} + x_{i-1})^+, \quad i = 2, \ldots, n, \tag{3.5}$$

$$L = (W_n + Z_n + \sum_{i=1}^{n-1} x_i - d)^+, \tag{3.6}$$

$$G = (-W_n - Z_n - \sum_{i=1}^{n-1} x_i + d)^+. \tag{3.7}$$

Note that $S_1 = W_1 = 0$ since (by assumption) the first job commences at $t = 0$. Note also that waiting and idling, and tardiness and earliness, satisfy a parity relationship, i.e., $W_i \cdot S_i = 0, i = 2, ..., n,$ and $L \cdot G = 0$.

Assuming linear costs for waiting, idling and tardiness, the appointments scheduling problem (ASP) is to find a schedule of times for customer arrivals that minimize the following function:

$$\min_{\mathbf{x}} \left\{ \sum_{i=2}^{n} c_i^w E[W_i] + \sum_{i=2}^{n} c_i^s E[S_i] + c_\ell E[L] \right\}, \tag{3.8}$$

where the expectations are over **Z**. The use of conventional non-linear optimization techniques for solving (3.8) is problematic when there are several jobs because evaluation of the objective function, and its gradient, necessitates the computation of multi-dimensional integrals which typically have no known closed-form expressions. Our approach is to use a stochastic linear programming formulation to overcome this difficulty. However, before we present the formulation we first discuss a case for which the ASP is easy to solve.

3.2.1 When is the ASP Easy?

The simplest form of (3.8) occurs when there are only two jobs ($n = 2$) and $c_\ell = 0$. Owing to the low-dimensionality of the problem it is possible to derive a closed form expression for the optimal job allowance for the first job (job allowance for the second job is immaterial). This problem was first identified as a variation of the newsvendor problem by Weiss (1990).

What follows in this section is therefore a direct consequence of Weiss' work. Letting $F_1(\cdot)$ denote the cumulative distribution function (c.d.f.) of the first job, and $\bar{F}_1 = 1 - F_1$, ASP can be written as follows:

$$\min_{x_1} \{c_2^w E[W_2] + c_2^s E[S_2]\}, \tag{3.9}$$

where

$$E[W_2] = \int_0^\infty (z_1 - x_1)^+ dP(z_1) = \int_{x_1}^\infty z_1 dF_1(z_1) - x_1 \bar{F}_1(x_1)$$

and

$$E[S_2] = \int_0^\infty (x_1 - z_1)^+ dP(z_1) = -\int_0^{x_1} z_1 dF_1(z_1) + x_1 F_1(x_1).$$

The objective function is easily shown to be convex and the optimal allowance for job 1 is obtained as the following critical fractile (analogous to the newsvendor solution):

$$x_1^* = F_1^{-1}\left\{\frac{c_2^w}{c_2^w + c_2^s}\right\}. \tag{3.10}$$

It follows from (3.10) that as the waiting (idling) cost increases the optimal job allowance increases (decreases). Furthermore, the distribution of the second job's duration plays no role in determining optimal x_1^*. It is possible to derive optimality conditions for optimal x_1^* and x_2^* when $n = 3$ (see, Robinson, Gerchak and Gupta, 1996, for details). However, the method becomes impractical for $n > 3$.

3.2.2 Stochastic Linear Program Formulation of the ASP

In this section, we formulate the ASP as a stochastic linear program. First, (3.8) is written as the following deterministic equivalent of a two-stage stochastic linear program (2S-SLP). (See Birge and Louveaux (1997), Kall and Wallace (1994), Dempster (1986), Ermoliev and Wets (1988), and references therein for more details.)

$$\min E\left\{\sum_{i=2}^{n} c_i^w w_i \quad + \quad \sum_{i=2}^{n} c_i^s s_i \quad + \quad c_\ell \ell\right\}$$

s.t.

$$\begin{array}{cccccccc} +w_2 & & & & -s_2 & & = & Z_1 - x_1 \\ -w_2 & +w_3 & & & & -s_3 & = & Z_2 - x_2 \\ & \ddots & \ddots & & & \ddots & & \vdots \\ & & -w_n & +\ell & & & -g & = Z_n - d + \sum_{j=1}^{n-1} x_j \end{array} \tag{3.11}$$

$$\mathbf{x} \geq 0,\ w_i \geq 0,\ s_i \geq 0\ \forall i = 1, ..., n,\ \text{and}\ \ell, g \geq 0.$$

The summations in the objective function of (3.11) begin with index 2 because W_1 and S_1 are set to zero, leaving $n-1$ decision variables, $(x_1, x_2, \ldots, x_{n-1})$, for an n job problem. The first stage decision variables, $\mathbf{x}$, and second stage decision variables, $\mathbf{w}$, $\mathbf{s}$, ℓ, g, are written in lower case and the dependence of second stage decisions on the random variables, $\mathbf{Z}$, is implied. The constraints in (3.11) enforce the piecewise linearity of the waiting, idling and tardiness functions. The first $n-1$ constraints correspond to waiting/idling time of customers/jobs 2 through n, and the n^{th} constraint corresponds to tardiness/earliness. We can rewrite the 2-SLP above more compactly as follows:

$$\min_{\mathbf{x}} \{Q(\mathbf{x})\} \tag{3.12}$$

where $Q(\mathbf{x}) = E[Q(\mathbf{x}, \mathbf{Z})]$ and

$$Q(\mathbf{x}, \mathbf{Z}) = \min_{\mathbf{y}} \{\mathbf{cy} \mid \mathcal{T}\mathbf{x} + \mathcal{W}\mathbf{y} = \mathbf{h},\ \mathbf{y} \geq 0\},\ \mathbf{c} = \begin{bmatrix} \mathbf{c}^w \\ \mathbf{c}^s \\ c_\ell \\ 0 \end{bmatrix},\ \mathbf{y} = \begin{bmatrix} \mathbf{w} \\ \mathbf{s} \\ \ell \\ g \end{bmatrix},\ \mathbf{h} = \begin{bmatrix} Z_1 \\ \vdots \\ Z_n - d \end{bmatrix}. \tag{3.13}$$

In the stochastic programming literature $Q(\mathbf{x})$ is called the *recourse function*, whereas $\mathcal{T}$ and $\mathcal{W}$ ($n \times n-1$ and $n \times 2n$ matrices, respectively) are called the *technology* and *recourse* matrices. We can write the recourse matrix for this problem as $\mathcal{W} = [\mathcal{W}' \mid -I]$ where I is the identity matrix. The matrix $\mathcal{T}$ and the submatrix $\mathcal{W}'$ have the following form (empty

spaces indicate zeroes):

$$\mathcal{T} = \begin{bmatrix} 1 & & \\ & \ddots & \\ & & 1 \\ -1 & \cdots & -1 \end{bmatrix}, \quad \mathcal{W}' = \begin{bmatrix} 1 & & & \\ -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \end{bmatrix}.$$

When viewed as a 2-SLP the two-job problem is a *simple recourse problem* which follows from the fact that the recourse matrix has the form $\mathcal{W} = [1 \mid -1]$. In general, problems where the recourse matrix is of the form $[I \mid -I]$ can be solved very efficiently due to separability of the second stage constraints. The multi-job problem (3.11) differs from this because of a sub-diagonal of -1s in $\mathcal{W}'$, resulting in a nonseparable second stage. However, the second stage is feasible for any $\mathbf{x} \in \Re^{n-1}$, i.e., $pos(W) = \Re^{n-1}$. Put differently, the 2-SLP has a *complete recourse*.

The dual of the ASP can be written as

$$\max\{E[\pi\mathbf{Z}] \mid E[\pi]\mathcal{T} \leq 0, \pi\mathcal{W} \leq \mathbf{c}\}. \tag{3.14}$$

Decomposition methods (e.g., Bender's decomposition, Dantzig-Wolfe decomposition), whether applied to the primal (3.12) or dual (3.14), rely on efficient solution of the *subproblems*, $Q(\mathbf{x}, \mathbf{Z})$. However, in order to solve the ASP efficiently, we need to focus instead on solution methods that take advantage of the structure of the primal problem, since in our case the solution of subproblems, defined by (3.13), is trivial. Notice that the latter requires only the evaluation of piecewise linear functions (waiting, idling and tardiness). Similarly, the dual of (3.13),

$$\max\left\{(\mathbf{Z} - \mathcal{T}\mathbf{x})\pi \mid \mathcal{W}^T\pi \leq \mathbf{c}\right\}, \tag{3.15}$$

can also be solved without actually solving a linear program. It has random cost coefficients but the constraints are fixed. Thus, the feasible region in (3.15) is the same for all $\mathbf{x}$ and $\mathbf{Z}$, and is easily shown to be compact. The optimal solution to (3.15) for a given choice of

$\mathbf{x}$ and realization of $\mathbf{Z}$ follows from the recursive structure of waiting, idling and tardiness functions. If there is nonzero waiting for jobs i and $(i+1)$ then any increase in job i's waiting time (i.e. an increase in the RHS of constraint i in 3.13) results in a subsequent increase in job $(i+1)$'s, and so on. On the other hand, if a job has nonzero idle time then it is decoupled from future ones because the next job starts at the scheduled time. As a result we can write the solution to (3.15) as the following backward recursion

$$\pi_i^*(\mathbf{x},\mathbf{Z}) = \begin{cases} -c_{i+1}^s & w_i = 0 \\ \\ c_{i+1}^w + \pi_{i+1}^*(\mathbf{x},\mathbf{Z}) & w_i > 0 \end{cases} ; \tag{3.16}$$

where

$$\pi_n^*(\mathbf{x},\mathbf{Z}) = \begin{cases} 0 & \ell = 0 \\ c_\ell & \ell > 0. \end{cases} \tag{3.17}$$

The dual solution, π, is the subgradient of $Q(\mathbf{x},\mathbf{Z})$ with respect to the RHS of the equality constraints in 3.13. It is continuous everywhere except on a set of points of measure zero. For example, $w_i = 0$ implies that $w_{i-1} + Z_{i-1} - x_{i-1} \leq 0$. When strict equality is satisfied, i.e., $w_i = s_i = 0$, there is degeneracy in the second stage problem and $\pi_i^*(\mathbf{x},\mathbf{Z}) \in [-c_{i+1}^s, c_{i+1}^w + \pi_{i+1}^*(\mathbf{x},\mathbf{Z})]$.

Before closing this section, we present a property of the ASP that allows us to establish an equivalence between two common cost structures.

Proposition 1 *Expected idle time is equal to the difference between two sums: the sum of expected tardiness and the session length, and the sum of average job durations and expected earliness, i.e.,*

$$\sum_{i=1}^{n} E[S_i] = [E[L] + d] - [E[G] + \sum_{i=1}^{n} \mu_i].$$

This is easily seen by adding the equality constraints in (3.11) and taking the expectation of both sides. If the session length d is zero then expected tardiness is equal to expected makespan (time for completion of all jobs) and expected earliness is zero. Having established

proposition 1, it is easy to see that the solution to the ASP remains unchanged under the following two conditions:

- idle time costs are identical, for example if $c_i^s = \alpha$, $\forall i$, and tardiness cost is zero, i.e. $c_\ell = 0$,
- idle time costs are zero for all jobs, i.e., $c_i^s = 0$ and $c_\ell = \alpha$.

In the two situations described above, the objective function differs only by a constant, i.e., the sum of the first moments of the job durations. Robinson et al. (1996) do not consider tardiness cost in their formulation, and use a similar argument to net out the sum of the first moments of job durations from the objective function.

3.3 Solution Method and Aggregation Bounds

When Ξ is finite and the number of scenarios is not computationally prohibitive methods such as the L-shaped algorithm can be used. Approximate methods for the cases in which Ξ is continuous or the number of discrete scenarios is prohibitively large are typically based on partitioning Ξ and solving the resulting large scale linear program. For example, statistical sampling is used to obtain a discrete set of scenarios that define an approximate problem, which is then solved using the L-shaped method (Dantzig and Glynn, 1990, Infanger, 1992). Alternatively, quasi-gradient methods (Ermoliev, 1988) use general purpose nonlinear optimization techniques where the objective and its gradient are obtained from sampled data. These methods rely on statistical estimates of the objective and its gradient for solution. In this section we provide an efficient approximation for computing the optimal job allowances and deterministic bounds on the accuracy loss due to the approximation.

3.3.1 Sequential Bounding Algorithm

If Ξ is finite, then (3.11) is a linear program with the *block-diagonal* structure. Each block corresponds to the evaluation of (second stage) waiting, idling and tardiness costs that result from a given (first stage) set of job allowances, $\mathbf{x}$, and realization of job durations, $\mathbf{z}$. We refer to a set of realizations of job durations, $\{z_1, z_2, ..., z_n\}$, as a scenario, ω_k, where $k = 1, ..., K$ indexes the K scenarios. Because of the simple form of the second stage problem in the ASP, decomposition algorithms are very efficient at solving large scale problems. However, if independent finite service time distributions are specified, the number of scenarios grows geometrically with respect to n, or alternatively, if service time distributions are continuous then the number of scenarios is infinite. In such cases an adaptation of the L-shaped algorithm based on partitioning the space of the random service durations can be used. We briefly review this below (see Chapter 5 of Birge and Louveaux, 1997) for a detailed review).

The basic idea of the L-shaped algorithm with sequential bounding (LSB) is to use constraints, based on *lower bounding functionals*, and upper bounds for $Q(\mathbf{x})$ to approximate the optimal solution, $\mathbf{x}^*$. Classic bounds such as the Jensen lower bound and the Edmundson-Madansky bound can be generalized over a partition of the support (Huang, Ziemba and Ben-Tal, 1977). For example, the Jensen bound can be written as

$$E[f(\mathbf{x}, \mathbf{Z})] \geq \sum_{k=1}^{\nu} p^k f(\mathbf{x}, \mathbf{z}^k)$$

where $f(\cdot)$ must be a convex function of the components of random vector $\mathbf{Z}$, k indexes cells of a partition of Ξ, p^k is the conditional probability on the k^k cell, and $\mathbf{z}^k$ is the vector of conditional expectations of the job durations on that cell. We denote the partition of Ξ by $S^{(\nu)}$ where $S^{(\nu)} = \{S^k,\ k = 1, ..., \nu\}$ and for simplicity we assume a rectangular partition, i.e.,

$$S^k = [a_1^{(k)}, b_1^{(k)}] \times [a_2^{(k)}, b_2^{(k)}] \times \cdots \times [a_n^{(k)}, b_n^{(k)}].$$

where n is the number of jobs. Thus if job durations are independent then the integrals for p^k and $\mathbf{z}^k$ are independent and can be iterated. When the partition is refined in such a way that the approximate distribution $\{p^k,\ k = 1, ..., \nu\}$ converges to the true distribution, P, the bound converges to the expectation of the function as $\nu \to \infty$.

We first outline the LSB and postpone the discussion of how to obtain the upper bounds used in the algorithm to the next section. In our description the lower bounding functionals used to outer linearize the recourse function are hyperplanes that are obtained using Jensen bounds. As such, the discrete approximation is

$$\begin{aligned} &\min \textstyle\sum_{k=1}^{\nu} p^k \sum_{i=2}^{n} \mathbf{c}\mathbf{y}^k \\ s.t.\ \ &\mathcal{T}\mathbf{x} + \mathcal{W}\mathbf{y}^k = \mathbf{h}^k,\ k = 1, ..., \nu, \\ &\mathbf{x} \geq 0,\ \mathbf{y}^k \geq 0, \quad k = 1, ..., \nu. \end{aligned} \tag{3.18}$$

We refer to the optimal solution of (3.18) as $(\mathbf{x}^{(\nu)}, (\mathbf{y}^{k*}, k = 1, ..., \nu))$ and its objective function at the optimum as $Q^{(\nu)}$, where $Q^{(\nu)} = \sum_{k=1}^{\nu} p^k Q(\mathbf{x}^{(\nu)}, \mathbf{z}^k) = \theta^{(\nu)}$. At each iteration the objective, $Q^{(\nu)}$, is a lower bound on the optimal solution. The basic form of the algorithm is as follows:

L-Shaped Algorithm with Sequential Bounding

step 1: Let ν index the iteration. Set $\nu = 0$.

step 2: Set $\nu = \nu + 1$. Solve the discrete problem (3.18) defined by partition $\mathcal{S}^{\nu}$ using the standard L-shaped method and let $(\mathbf{x}^{(\nu)}, \theta^{(\nu)})$ be the optimal solution.

step 3: Evaluate $Q^{UB}(\mathbf{x}^{(\nu)})$. If $Q^{UB}(\mathbf{x}^{(\nu)}) - \theta^{(\nu)} \leq$ *tolerance* then stop. Otherwise go to step 4.

step 4: Refine the current partition $S^{(\nu)} \to S^{(\nu+1)}$ and return to step 2.

Note that the stopping criterion in *step 3* is based on the absolute difference between the upper and lower bounds, and this places a limit on the accuracy loss due to solving the discrete approximation.

The above algorithm generates bounds on the gap between the solution obtained from solving the discrete version of the ASP at each step and the optimal solution. The gap depends on how the partition is refined in step 4 at each iteration. If the partition is refined in such a way that the discrete distribution converges to the true distribution as $\nu \to \infty$, then $\mathbf{x}^{(\nu)} \to \mathbf{x}^*$ (Birge and Wets, 1986).

Refinement of a partition is typically described as involving three decisions. The first is the choice of a cell to split, k^*, the second is the direction along which to make the split, i^*, and the third is the point at which to make the split, $c_{i^*}^{k^*}$. After the split is made the old and new cells, S^{k^*} and $S^{\nu+1}$ respectively, are

$$S^{k^*} = [a_1^{k^*}, b_1^{k^*}] \times [a_2^{k^*}, b_2^{k^*}] \times \cdots \times [a_{i^*}^{k^*}, c_{i^*}^{k^*}] \times \cdots \times [a_n^{k^*}, b_n^{k^*}] \tag{3.19}$$

$$S^{\nu+1} = [a_1^{k^*}, b_1^{k^*}] \times [a_2^{k^*}, b_2^{k^*}] \times \cdots \times [c_{i^*}^{k^*}, b_{i^*}^{k^*}] \times \cdots \times [a_n^{k^*}, b_n^{k^*}]. \tag{3.20}$$

The aim is to obtain solutions to the approximate problem that converge to the optimum quickly. We describe a simplified version of the method proposed by Frauendorfer and Kall (1988) (which is suitable to guarantee convergence of the probability distribution in the limit). Choose the cell, k^*, which has the largest difference between the upper and lower bound, i.e.,

$$k^* = \text{argmin}_{k=1,\ldots,\nu}\{Q^{k,UB}(\mathbf{x}^{(\nu)}) - Q^{k,LB}(\mathbf{x}^{(\nu)})\}. \tag{3.21}$$

Note that the upper and lower bounds in (3.21) are for a particular cell. For example, the conditional Jensen bound on a given cell, k, is $Q^{k,LB}(\mathbf{x}^{(\nu)}) = p^k Q(\mathbf{x}^{(\nu)}, \mathbf{z}^k)$. The rationale behind choosing the cell with the largest difference is that it has the highest potential for improvement. In choosing the direction, i^*, it is desirable to choose one along which there is a high degree of nonlinearity of the recourse function. Along a given direction i evaluate

$$\epsilon_1^i = Q(\mathbf{x}^{(\nu)}, \mathbf{v}^2) - Q(\mathbf{x}^{(\nu)}, \mathbf{v}^1) - \pi(\mathbf{z}^k, \mathbf{v}^1)(\mathbf{v}^2 - \mathbf{v}^1), \tag{3.22}$$

$$\epsilon_2^i = Q(\mathbf{x}^{(\nu)}, \mathbf{v}^1) - Q(\mathbf{x}^{(\nu)}, \mathbf{v}^2) - \pi(\mathbf{z}^k, \mathbf{v}^2)(\mathbf{v}^1 - \mathbf{v}^2) \tag{3.23}$$

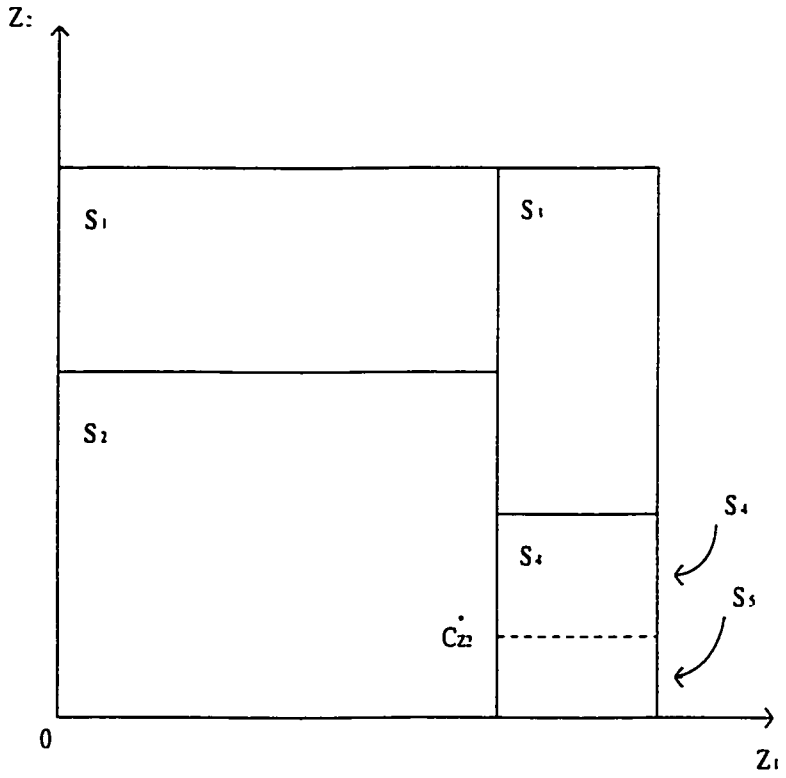

Figure 3.1: Illustration of the method for choosing a cell from partition Ξ.

where $(\mathbf{v}^1, \mathbf{v}^2)$ is a pair of adjacent vertices of S^k, $\mathbf{v}^1 = (a_1^k, ..., a_i^k, ..., a_n^k)$, $\mathbf{v}^2 = (a_1^k, ..., b_i^k, ..., a_n^k)$. From the subgradient inequality, ϵ_1 and ϵ_2 are nonnegative since $Q(\mathbf{x}^{(\nu)}, \mathbf{z})$ is convex. The direction is chosen such that $i^* = \text{argmax}\{\min\{\epsilon_1^i, \epsilon_2^i\}\}$. The point at which to split, c_{i^*}, is then chosen so that

$$Q(\mathbf{x}^{(\nu)}, \mathbf{v}^2) + \pi_{i^*}(\mathbf{z}^k, \mathbf{v}^2)(c_{i^*} - b_{i^*}) = Q(\mathbf{x}^{(\nu)}, \mathbf{v}^1) + \pi_{i^*}(\mathbf{z}^k, \mathbf{v}^1)(c_{i^*} - a_{i^*}).$$

The choice of split point for a two dimensional example is illustrated in Figure 3.2.

3.3.2 Aggregation Bounds

Standard methods for obtaining upper bounds of convex expectational functionals rely on determining an approximate discrete distribution such that its support is composed of

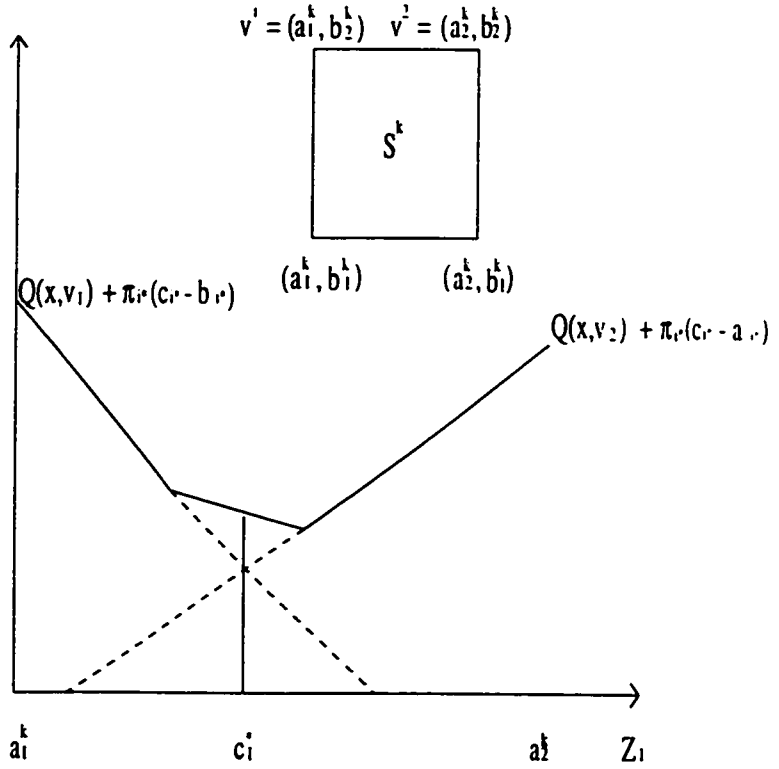

Figure 3.2: Illustration of the method for choosing the split point for cell k^* and direction i^*.

extreme points, of Ξ. For example, the Edmundson-Madansky upper bound is a weighted average of the function at the extreme points and thus when Ξ is not compact the bound is not defined. It requires evaluation of $Q(\mathbf{x}, \mathbf{z})$ at each vertex of each cell in the partition. Since the number of vertices increases geometrically with the number of dimensions the computation time quickly becomes prohibitive as the number of random variables increases. A method based on solving a generalized moment problem, which is applicable to the case in which Ξ is not compact, was developed by Birge and Wets (1995), and was subsequently applied to the problem of computing upper bounds on tardiness in a project network (Birge and Maddox, 1995). However, it is not well suited for use in an optimization algorithm.

We now show how efficient upper bounds can be obtained that are independent of service

duration distributions by using a dual representation of the ASP. Using (3.13) we can write the recourse function in the following form

$$Q(\mathbf{x}) = \int_{\Xi} \pi(\mathbf{x}, \mathbf{z})(\mathbf{h} - \mathcal{T}\mathbf{x}) dP(\mathbf{z}). \tag{3.24}$$

Of course it is generally not possible to compute the integral in (3.24) exactly. However, we can obtain an upper bound on (3.24) using an approach similar to the aggregation bounds described by Zipkin (1979) for deterministic linear programs, and subsequently extended by Birge (1985) to the case of multi-stage stochastic linear programs. As before we let $S^{\nu} = \{S^k,\ k = 1, ..., \nu\}$ denote a rectangular partition of the support, Ξ, where the $\mathbf{h}^k$ are conditional expectations, and the p^k are conditional probabilities. We start with an application of aggregation bounds to the ASP.

Proposition 2

$$Q^{(\nu)} \leq Q(\mathbf{x}^*) \leq Q^{(\nu)} + \epsilon_1(\nu)$$

where $Q^{(\nu)} = \sum_{k=1}^{\nu} p^k \mathbf{c}\mathbf{y}^{k*}$ *and*

$$\epsilon_1(\nu) = \sum_{k=1}^{\nu} \sum_{i=2}^{n} \int_{S^k} ((\sum_{j=i}^{n} c_j^w + c_\ell)(h_i - h_i^k)^+ + c_i^s (h_i^k - h_i)^+) dP(\mathbf{z}). \tag{3.25}$$

Proof: The lower bound follows from the fact that the objective in (3.18) is a Jensen bound for any $\mathbf{x}$ and hence the optimal solution to (3.18) is a lower bound on the optimal solution to the ASP. The second inequality can be proved in the following way:

$$\begin{aligned} Q(\mathbf{x}^*) &= \int_{\Xi} \pi(\mathbf{z}, \mathbf{x}^*) \mathbf{h} dP(\mathbf{z}) && (3.26) \\ &\leq \int_{\Xi} \pi(\mathbf{z}, \mathbf{x}^*) \mathbf{h} dP(\mathbf{z}) \ + && (3.27) \\ &\qquad \sum_{k=1}^{\nu} \int_{S^k} ((\mathbf{c} - \pi(\mathbf{z}, \mathbf{x}^*) W) \mathbf{y}^{k*} - \pi(\mathbf{z}, \mathbf{x}^*) \mathcal{T} \mathbf{x}^{(\nu)}) dP(\mathbf{z}) \end{aligned}$$

where the inequality in (3.27) is due to nonnegativity of the second term which results from the dual constraints in (3.14). Reorganizing the terms we can write the right hand side as

follows:

$$= Q^{(\nu)} + \sum_{k=1}^{\nu} \int_{S^k} \pi(\mathbf{z}, \mathbf{x}^*)(\mathbf{h} - \mathcal{W}\mathbf{y}^{k*} - \mathcal{T}\mathbf{x}^{(\nu)}) dP(\mathbf{z}) \tag{3.28}$$

$$\leq Q^{(\nu)} + \sum_{k=1}^{\nu} \sum_{i=2}^{n} \int_{S^k} (\pi_i^{UB}(h_i - \mathcal{W}_{(i\cdot)}\mathbf{y}^{k*} - \mathcal{T}\mathbf{x}^{(\nu)})^+ - \pi_i^{LB}(\mathcal{W}_{(i\cdot)}\mathbf{y}^{k*} + \mathcal{T}\mathbf{x}^{(\nu)} - h_i)^+) dP(\mathbf{z}) \tag{3.29}$$

$$= Q^{(\nu)} + \sum_{k=1}^{\nu} \sum_{i=2}^{n} \int_{S^k} (\pi_i^{UB}(h_i - h_i^k)^+ - \pi_i^{LB}(h_i^k - h_i)^+) dP(\mathbf{z}) \tag{3.30}$$

where π_i^{UB} and π_i^{LB} are upper and lower bounds on the dual solution for any $\mathbf{x} \in \Re^{n-1}$ and $\mathbf{Z} \in \Xi$ and the positive and negative parts of the integrand in (3.29) have been separated to give an overall upper bound on $Q(\mathbf{x}^*)$. The feasible region of the dual is compact and the bounds can be obtained from (3.16) and (3.17) as follows

$$\pi_i^{UB} = \max\{\pi_i(\mathbf{x}, \mathbf{Z}) \mid \mathbf{x} \in \Re^{n-1}, \mathbf{Z} \in \Xi\} \tag{3.31}$$

$$= c_{i+1}^w + \max\{\pi_{i+1}(\mathbf{x}, \mathbf{Z}) \mid \mathbf{x} \in \Re^{n-1}, \mathbf{Z} \in \Xi\} \tag{3.32}$$

$$= \sum_{j=i+1}^{n} c_j^w + c_\ell, \tag{3.33}$$

and

$$\pi_i^{LB} = \min\{\pi_i(\mathbf{x}, \mathbf{Z}) \mid \mathbf{x} \in \Re^{n-1}, \mathbf{Z} \in \Xi\} = -c_{i+1}^s. \tag{3.34}$$

□

The bounds in proposition 2 are on the optimal solution to the ASP and do not necessarily provide any information about the the recourse function at $\mathbf{x}^{(\nu)}$, $Q(\mathbf{x}^{(\nu)})$. To bound the accuracy of the LSB algorithm we need bounds on the recourse function at the current iterate $\mathbf{x}^{(\nu)}$. We now derive a similar bound on $Q(\mathbf{x}^{(\nu)})$.

Proposition 3

$$Q^{(\nu)} \leq Q(\mathbf{x}^{(\nu)}) \leq Q^{(\nu)} + \epsilon_2(\nu) \; . \tag{3.35}$$

where

$$\epsilon_2(\nu) = \sum_{k=1}^{\nu}\sum_{i=2}^{n}\int_{S^k}(\pi_i^{k,UB}(h_i - h_i^k)^+ - \pi_i^{k,LB}(h_i^k - h_i)^+)dP(\mathbf{z})$$

and $\pi_i^{k,UB}(\mathbf{x}^{(\nu)})$, $\pi_i^{k,LB}(\mathbf{x}^{(\nu)})$ *are upper and lower bounds, given* $\mathbf{x}^{(\nu)}$ *and* $\mathbf{Z} \in S^k$.

Proof: The lower bound in (3.35) follows from the fact that $Q^{(\nu)}$ is a lower bound on the optimum (proposition 2). The upper bound is proved as follows:

$$Q(\mathbf{x}^{(\nu)}) = \int_{\Xi}\pi(\mathbf{z},\mathbf{x}^{(\nu)})(\mathbf{h} - \mathcal{T}\mathbf{x}^{(\nu)})dP(\mathbf{z}) \tag{3.36}$$

$$\leq \int_{\Xi}\pi(\mathbf{z},\mathbf{x}^{(\nu)})(\mathbf{h} - \mathcal{T}\mathbf{x}^{(\nu)})dP(\mathbf{z}) + \sum_{k=1}^{\nu}\int_{S^k}(\mathbf{c} - \pi(\mathbf{z},\mathbf{x}^{(\nu)})\mathcal{W})\mathbf{y}^{k*}dP(\mathbf{z}) \tag{3.37}$$

where the inequality follows from the dual constraints on the discrete approximation (3.18). Rearranging the terms as in proposition 2

$$Q(\mathbf{x}^{(\nu)}) \leq Q^{(\nu)} + \sum_{k=1}^{\nu}\sum_{i=2}^{n}\int_{S^k}(\pi_i^{k,UB}(h_i - h_i^k)^+ - \pi_i^{k,LB}(h_i^k - h_i)^+)dP(\mathbf{z}) \tag{3.38}$$

where $\pi_i^{k,UB} = \max\{\pi_i(\mathbf{x}^{(\nu)},\mathbf{Z}) \mid \mathbf{Z} \in S^k\}$ and $\pi_i^{k,LB} = \min\{\pi_i(\mathbf{x}^{(\nu)},\mathbf{Z}) \mid \mathbf{Z} \in S^k\}$. □

The lower bounds in propositions 2 and 3 are identical. That the upper bound in proposition 3 is tighter than the bound in proposition 2 follows from the fact that

$$\pi_i^{k,UB} \leq \pi_i^{UB} \text{ and } \pi_i^{k,LB} \geq \pi_i^{LB},\ k = 1,...,\nu.$$

The upper bound in proposition 3 is an upper bound on $Q(\mathbf{x}^{\nu})$, and hence on $Q(\mathbf{x}^*)$ as well. Thus from propositions 2 and 3 the solution to (3.18) provides upper and lower bounds on the accuracy loss due to the approximation, i.e., $|Q(\mathbf{x}^*) - Q(\mathbf{x}^{(\nu)})| \leq \epsilon_2(\nu)$. It remains to be shown how upper and lower bounds for the dual solution on a partition of Ξ can be obtained. The following simple dynamic programming procedure can be used.

Bounding the Dual Multipliers on a Cell:

For each job we must determine whether nonzero waiting or idling is possible and whether nonzero tardiness or earliness is possible for $\mathbf{x}^{(\nu)}$ and $\mathbf{Z} \in S^k$. From (3.4) and (3.6) waiting

times and tardiness are nondecreasing in $\mathbf{Z}$ and hence we can bound them by evaluating them at the extreme points of the cell as follows:

$$W_i^{k,UB} = (W_{i-1}^{k,UB} + b_{i-1}^k - x_{i-1})^+, \ i = 2, \ldots, n, \tag{3.39}$$

$$W_i^{k,LB} = (W_{i-1}^{k,LB} + a_{i-1}^k - x_{i-1})^+, \ i = 2, \ldots, n, \tag{3.40}$$

and

$$L^{k,UB} = (W_n^{k,UB} + b_n^k + \textstyle\sum_{i=1}^n x_i - d)^+, \ i = 2. \ldots, n, \tag{3.41}$$

$$L^{k,LB} = (W_n^{k,LB} + a_n^k + \textstyle\sum_{i=1}^n x_i - d)^+, \ i = 2, \ldots, n. \tag{3.42}$$

Note that when Ξ is not compact the upper bounds in (3.39) and (3.41) may be infinite. However, we are concerned only with knowing whether they are greater than zero. A nonzero upper bound on waiting time indicates a lower bound of zero on the corresponding idle time, and vice versa due to the parity relationship. The same is true for tardiness and earliness. The upper and lower bounds on the dual solution can therefore be obtained using the following backward recursions:

$$\pi_i^{k,UB}(\mathbf{x}) = \begin{cases} -c_{i+1}^s & \text{if } W_{i+1}^{k,UB} = 0 \\ \max\{-c_{i+1}^s, c_{i+1}^w + \pi_{i+1}^{k,UB}(\mathbf{x})\} & \text{if } W_{i+1}^{k,LB} = 0 \text{ and } W_{i+1}^{k,UB} > 0 \\ c_{i+1}^w + \pi_{i+1}^{k,UB}(\mathbf{x}) & \text{if } W_{i+1}^{k,LB} > 0 \end{cases} \tag{3.43}$$

and

$$\pi_i^{k,LB}(\mathbf{x}) = \begin{cases} -c_{i+1}^s & \text{if } W_{i+1}^{k,UB} = 0 \\ \min\{-c_{i+1}^s, c_{i+1}^w + \pi_{i+1}^{k,LB}(\mathbf{x})\} & \text{if } W_{i+1}^{k,LB} = 0 \text{ and } W_i^{k,UB} > 0 \\ c_{i+1}^w + \pi_{i+1}^{k,LB}(\mathbf{x}) & \text{if } W_{i+1}^{k,LB} > 0 \end{cases} \tag{3.44}$$

for $i = 1, ..., n-1$, and the upper and lower bounds for π_n are

$$\pi_n^{k,UB}(\mathbf{x}) = \begin{cases} 0 & \text{if } L^{k,UB} = 0 \\ c_\ell & \text{if } L^{k,UB} > 0 \end{cases} \tag{3.45}$$

and

$$\pi_n^{k,LB}(\mathbf{x}) = \begin{cases} 0 & \text{if } L^{k,LB} = 0 \\ c_\ell & \text{if } L^{k,LB} > 0. \end{cases} \tag{3.46}$$

The bounds in proposition 7 can be viewed as having a penalty term on the discrepancy between the approximate discrete problem and the continuous problem. The penalty for a given cell is expressed as $E_{S^k}[\pi_i^{k,UB}(h_i - h_i^k)^+ - \pi_i^{k,LB}(h_i^k - h_i)^+]$ above. When it is used in LSB, refinement of the partition results in a reduction of this measure of discrepancy on the chosen cell at $\mathbf{x}^{(\nu)}$.

3.4 Approximations and Heuristics

As the number of jobs to be scheduled increases the problem size grows, and the corresponding computational effort in obtaining an effective partition of Ξ that suitably approximates the true problem increases. In this section we point out two relaxations that give lower bounds on $Q(\mathbf{x}^*)$ and significantly reduce computation time. The first approximation is based on a relaxation of the n job problem to obtain a set of separable subproblems.

Proposition 4 *If $c_i^s = c_j^s$, $\forall(i,j)$ then the simple recourse problem,*

$$\min_{\mathbf{x}}\{E_{\mathbf{Z}}[\min\{\mathbf{cy} \mid \mathcal{T}\mathbf{x} + [I \mid -I]\mathbf{y} = \mathbf{h}\}]\}, \tag{3.47}$$

gives a lower bound on the ASP.

Proof: Let $c_i^s = c^s$, $\forall i$. We can show the following problem is equivalent to the ASP

$$\min E\left\{\sum_{i=2}^{n} c_i^w w_i \quad + \quad c^s s \quad + \quad c_\ell \ell\right\}$$

$s.t.$

$$w_2 - s_2 = Z_1 - x_1 \tag{3.48}$$

$$-w_2' + w_3 - s_3 = Z_2 - x_2 \tag{3.49}$$

$$\vdots$$

$$-w_{n-1}' + w_n - s_3 = Z_{n-1} - x_{n-1} \tag{3.50}$$

$$-w_n' + l - g = Z_n + \sum_{j=1}^{n-1} x_j - d \tag{3.51}$$

$$-w_n' + s = Z_n + \sum_{j=1}^{n-1} x_j \tag{3.52}$$

$$w_i = w_i', \; i = 2, ..., n. \tag{3.53}$$

$$\mathbf{x} \geq 0, \; w_i \geq 0, \; s_i \geq 0, \; \forall i = 1, ..., n, \text{ and } \ell, g, s \geq 0.$$

Note that s is total makespan. By proposition 1 it follows that $\sum_{i=1}^{n} s_i = s - \sum_{i=1}^{n} \mu_i$. Removing the constraint set (3.53) yields a relaxation of the ASP. Since the objective function is independent of $(w_i', \; s_i, \; \forall i)$ the pairs $(w_i', \; s_i)$ for $i = 1, ..., n-1$ in constraints (3.48) - (3.50) can be aggregated into one decision variable $s_i' = w_i' + s_i$. Since $Z_n + \sum_{j=1}^{n-1} x_j \geq 0$ the optimal solution has $w_n' = 0$. Removing w_n' from (3.51) and (3.52) yields a problem equivalent to (3.47). □

The above approximation achieves a complete separation of the n jobs. Less severe separations lead to improved lower bounds. For example, removing only constraint $i = m$ of (3.53) separates the problem into independent sequences of m and $n - m + 1$ job problems. (Note that the assumption that the first customer in a sequence does not wait implies the second sequence of jobs has $n - m + 1$ jobs rather than $n - m$ jobs.) For the special case of a complete separation of jobs and $c_\ell = 0$, (3.47) separates into a set of n independent 2-job (newsvendor) problems. If we assume job durations are independent, with distribution function $F_i(\cdot)$ for each job i, this yields the following solution to the relaxed problem

$$x_i = F_i^{-1}\left\{\frac{c_{i+1}^w}{c_{i+1}^w + c^s}\right\} \; \forall i.$$

Henceforth we refer to the relaxation based on separating jobs as the *Separation Heuristic* (SEP).

In the next approximation we utilize the Jensen bound and the recursive structure of the waiting times in (3.4). First, we define the following approximate expected waiting, idling and tardiness times:

$$E[\tilde{W}_i] = E[(E[\tilde{W}_{i-1}] + Z_{i-1} - x_{i-1})^+],\ i = 2, ..., n, \tag{3.54}$$

$$E[\tilde{S}_i] = E[(-E[\tilde{W}_{i-1}] - Z_{i-1} + x_{i-1})^+],\ i = 2, ..., n, \tag{3.55}$$

$$E[\tilde{L}] = E[(E[\tilde{W}_n] + Z_n + \sum_{i=1}^{n-1} x_i - d)^+]. \tag{3.56}$$

From the Jensen bound, and convexity of W_i with respect to W_{i-1}, we have

$$E[W_i] \geq E[\tilde{W}_i],\ E[S_i] \geq E[\tilde{S}_i],\ E[L] \geq E[\tilde{L}].$$

The structure of $E[\tilde{W}_i]$, $E[\tilde{S}_i]$ and $E[\tilde{L}]$ allows for efficient computation because the multi-dimensional integrals can be separated into one dimensional integrals. Thus we could solve

$$\min_{\mathbf{x}}\{\sum_{i=1}^{n}(c_i^w E[\tilde{W}_i] + c_i^s E[\tilde{S}_i]) + c_\ell E[\tilde{L}]\} \tag{3.57}$$

using standard nonlinear optimization to obtain a lower bound on $Q(\mathbf{x}^*)$. Again, for the case in which $c_\ell = 0$, it is straightforward to show that the optimal solution is the following solution to a modified newsvendor problem, which we refer to as the *Sequential Jensen Bound Heuristic* (SJB)

SJB Heuristic: *Set job allowances such that* $x_i = E[\tilde{W}_i] + F_i^{-1}\left\{\frac{c_{i+1}^w}{c_{i+1}^w + c_{i+1}^s}\right\}\ \forall i.$

The SJB heuristic provides a very simple decision rule, as well as lower bounds on the optimal solution. In the next section we offer some numerical examples that illustrate the performance of the approximations discussed in this section.

3.5 Insights and Examples

In addition to appointment scheduling, a related problem is to determine the optimal sequence in which jobs should be processed. This is relevant in those situations where jobs can be reordered prior to processing. In general, optimal sequencing is a hard problem due to its combinatorial nature. We begin this section by showing that the bounds obtained in the previous section can also be used to assess the potential for improvement through resequencing. For this purpose, we let $Q(\mathbf{x}^{os})$ and $Q(\mathbf{x}^{a})$ be the recourse function at the optimal solution of the ASP for the optimal sequence of jobs and some arbitrary sequence of jobs, respectively.

Proposition 5 *The optimal solution to the ASP for an arbitrary sequence is bounded by*

$$c_\ell(\sum_{i=1}^{n}\mu_i - d)^+ \leq Q(\mathbf{x}^a) \leq c_\ell(\sum_{i=1}^{n}\mu_i - d)^+ + \epsilon_1(1) \tag{3.58}$$

and the relative improvement due to resequencing is nonincreasing in c_ℓ.

Proof: Equation (3.58) follows directly from proposition 4 for the case $\nu = 1$. The bounds are obtained by solving a deterministic problem in this case ($p^1 = 1$, $\mathbf{h}^1 = (\mu_1, ..., \mu_{n-1}, \mu_n - d)$), and therefore are independent of job sequence. Defining the relative difference between the objective function for the optimal sequence and an arbitrary sequence as the ratio of the difference between the upper and lower bounds and their sum, we have

$$\frac{Q(\mathbf{x}^a) - Q(\mathbf{x}^{os})}{Q(\mathbf{x}^a) + Q(\mathbf{x}^{os})} \leq \frac{\epsilon_1(1)}{2c_\ell(\sum_{i=1}^{n}\mu_i - d)^+ + \epsilon_1(1)} \tag{3.59}$$

where, from (3.25), $\epsilon_1(1)$ is linearly increasing in c_ℓ. Thus the numerator and denominator in the left hand side of (3.59) are both linearly increasing in c_ℓ with the rate of increase of the denominator greater than or equal to the numerator. The relative improvement is therefore nonincreasing in c_ℓ. □

From proposition 5 it follows that as the cost of makespan and/or idle time increases, the

bounds on the relative benefits from optimizing job sequence are nonincreasing. Furthermore, if $\sum_{i=1}^{n} \mu_i > d$, then the bound on the relative benefits approaches zero as $c_\ell \to \infty$.

It can be shown for the case of $c_\ell = 0$ that some simple transformations of the random job durations in the ASP result in linear transformation of $\mathbf{x}^*$ and $Q(\mathbf{x}^*)$. For instance, consider changing each job duration by a constant factor $\mathbf{Z} \mapsto \mathbf{Z} + \mathbf{b}$ where $\mathbf{b} \in \Re^n$. It is easy to show that the effect on the optimal solution of the ASP in this case is $\mathbf{x}^* \mapsto \mathbf{x}^* + \mathbf{b}$ and $Q(\mathbf{x}^*)$ is unchanged. For the case of i.i.d. job durations and $c_\ell = 0$, the following can also be shown

Proposition 6 *The effect of the transformation* $\mathbf{Z} \mapsto a\mathbf{Z} + \mathbf{b}$, *where* $a \in \Re$ *and* $\mathbf{b} \in \Re^n$, *on the optimal solution, is* $Q(\mathbf{x}^*) \mapsto aQ(\mathbf{x}^*)$ and $\mathbf{x}^* \mapsto a\mathbf{x}^* + \mathbf{b}$

Proof: The proof follows from the fact that

$$\min_{\mathbf{y}} \{\mathbf{cy} \mid \mathcal{T}\mathbf{x} + \mathcal{W}\mathbf{y} = a\mathbf{h} + \mathbf{b},\ \mathbf{y} \geq 0\}$$

is equivalent to

$$\min_{\mathbf{y}} \left\{a\mathbf{c}\mathbf{y}' \mid \mathcal{T}\mathbf{x}' + \mathcal{W}\mathbf{y}' = \mathbf{h},\ \mathbf{y}' \geq 0\right\}$$

where $\mathbf{y}' = a^{-1}\mathbf{y}$ and $\mathbf{x}' = a^{-1}\mathbf{x} - a^{-1}\mathbf{b}$. □

From proposition 6 it follows that under the mean preserving transformation, $\mathbf{Z} \mapsto a\mathbf{Z} - (a-1)\mu$, $Q(\mathbf{x}^*)$ is increasing linearly with respect to the standard deviation of job durations. Also, if the solution for a particular a and $\mathbf{b}$ is known then the solution for any other a and $\mathbf{b}$ can be obtained by a simple transformation. Note also that several distributions have the property that they can be completely described through the above linear transformations (e.g. normal, uniform, exponential). Thus solving an instance of the problem in such cases results in knowing the solution for the entire class of problems by a trivial transformation of the solution.

3.5.1 Experimental Design

The numerical results presented below fall into three categories: (a) numerical experiments that give general insights and illustrate the quality of the aggregation bounds, (b) examples that reinforce the fact that the OR scheduling problem is economically important, and (c) numerical evidence that the heuristics work well for large n. The partitioning method described in section 3.3 was used to compute the solutions and deterministic bounds. In all cases 50 iterations were carried out, and 500 additional cells were added at each iteration. Solution times for the largest examples are typically less than 10 minutes on a modest workstation (Sun Ultra 10 with 128 MB Ram) for the largest problems considered. The master problem was solved using Cplex 5.0 at each iteration and the majority of computation time is spent in updating the partition. To simplify comparison of results the job distributions are assumed i.i.d. in each case. For each example we also compute a statistical estimate of the recourse function at the approximate solution, $\bar{Q}(\mathbf{x}^{(\nu)})$. (Since the ASP is a convex minimization problem these are statistical upper bounds on $Q(\mathbf{x}^*)$.) The statistical estimates were obtained using a sample size of 10^4, which is consistent with results of the simulation study by Ho and Lau (1992) that indicate an accuracy of $\pm 1\%$ at the 95% confidence level for $n \leq 30$.

3.5.2 Computation Times, Accuracy of LSB, and Parametric Variations

We start by providing some examples of the dependence of computation time on the number of jobs, n, and the number of cells in the partition, K. In other words, the results illustrate typical computation times for a single iteration of the LSB algorithm with given partition K. Results in Table 3.1 are examples with $n = 5$ to $n = 25$ and $K = 1000$ to $K = 50000$. Results are averages for 25 randomly generated problem instances in which cost parameters and job duration scenarios were varied. The job durations were assumed i.i.d uniformly distributed and were sampled according $U(0, 1)$ for each of the K scenarios. The cost coefficients for each

test problem were generated by sampling according to $U(0,1)$ for each job. Computation times were found to be relatively insensitive to changes in cost parameters.

Table 3.1 gives estimates of the average complexity per iteration of the LSB algorithm with respect to the number of jobs, n, and the number of cells, K. The complete algorithm requires other operations in addition to the solution of the discrete stochastic linear program approximation. At each iteration upper and lower bounds for each cell of the partition must be computed. Lower bounds are obtained from solution of the discrete problem; however, upper bounds must be computed for each cell. This is done by (a) computing upper and lower bounds on the dual multipliers (equations (3.43) - (3.46)) which are linear in n, and (b) computing conditional moments along each direction, which are linear in n. Given these bounds, selection of a cell for refinement requires a search for the cell with maximum gap which is also linear in n.

K/n	5	10	15	20	25
1000	0.42(61)	5.47(233)	28.54(438)	143.34(848)	337.91(1119)
5000	0.80(68)	7.85(229)	45.24(509)	151.24(820)	413.55(1244)
10000	1.54(67)	12.78(228)	66.17(512)	199.24(840)	505.34(1219)
50000	5.55(68)	39.56(233)	144.30(496)	405.19(877)	950.23(1346)

Table 3.1: Average computation times for various problem sizes in seconds (numbers in brackets are average numbers of iterations of the L-shaped algorithm).

Tables 3.2 and 3.3 contain results for problems with a variety of different cost structures. In each example we assume waiting and idling cost coefficients are the same for each job ($c_i^w = c_j^w$ and $c_i^s = c_j^s$, $\forall i, j$). In Table 3.2 there are no overtime costs, i.e. $c_\ell = 0$, and the relative costs of waiting and idling are varied. In Table 3.3 results for nonzero overtime costs are reported when the session length is assumed to be equal to the sum of the mean job durations ($d = 7.0$). The uniform distribution was chosen as a test case because it represents an extreme condition with respect to the application of a partitioning method

since the probability mass is evenly spread across the range of the distribution. In other words, the paritioning method cannot benefit from concentrating the partitioning within a smal region of high probability mass. Note that proposition 6 shows how the results in Table 3.2 can be transformed to correspond to any mean and variance of the job durations.

(c^s, c^w)	(9,1)	(8,2)	(7,3)	(6,4)	(5,5)	(4,6)	(3,7)	(2,8)	(1,9)
x_1	0.360	0.624	0.838	1.035	1.165	1.313	1.461	1.631	1.807
x_2	0.876	1.093	1.162	1.259	1.349	1.437	1.549	1.669	1.817
x_3	0.969	1.070	1.201	1.255	1.361	1.446	1.552	1.670	1.819
x_4	0.952	1.065	1.174	1.255	1.351	1.443	1.543	1.670	1.819
x_5	0.911	1.060	1.125	1.228	1.300	1.426	1.528	1.665	1.821
x_6	0.784	0.871	0.970	1.087	1.203	1.344	1.479	1.639	1.813
$Q(\mathbf{x})^{LB}$	8.963	13.423	15.726	16.656	16.400	15.139	12.906	9.674	5.394
$Q(\mathbf{x})^{UB}$	9.985	14.619	17.326	17.789	17.259	15.812	13.345	9.896	5.501
$\bar{Q}(\mathbf{x})$	9.077	13.498	15.855	16.858	16.551	15.278	12.983	9.768	5.412

Table 3.2: Results for 7 jobs with $U(0,2)$ job durations after 50 iterations with no tardiness cost penalty.

(c_ℓ, c^s, c^w)	(7,7,3)	(7,5,5)	(7,3,7)	(5,7,3)	(5,5,5)	(5,3,7)	(3,7,3)	(3,5,5)	(3,3,7)
x_1	0.606	0.831	1.063	0.645	0.875	1.136	0.719	0.997	1.250
x_2	1.085	1.175	1.267	1.113	1.217	1.337	1.125	1.250	1.375
x_3	1.080	1.197	1.264	1.106	1.236	1.308	1.120	1.250	1.375
x_4	1.091	1.196	1.266	1.125	1.216	1.321	1.131	1.251	1.383
x_5	1.067	1.104	1.208	1.049	1.137	1.252	1.077	1.194	1.351
x_6	0.936	0.997	1.164	0.956	1.009	1.203	0.935	1.069	1.242
$Q(\mathbf{x})^{LB}$	22.167	26.546	28.236	20.486	24.225	24.829	18.716	21.507	20.812
$Q(\mathbf{x})^{UB}$	25.045	29.114	30.699	22.754	26.457	26.438	20.547	23.282	21.891
$\bar{Q}(\mathbf{x})$	22.743	27.047	28.888	20.801	24.644	25.258	18.928	21.853	20.921

Table 3.3: Results for 7 jobs with $U(0,2)$ job durations after 50 iterations with tardiness cost penalty.

The results in Tables 3.2 and 3.3 illustrate a common behavior of solutions to the ASP for i.i.d. job durations, which is that they tend to have a dome shape, i.e., initially increasing and then decreasing job allowances. Numerical experiments for other types of distributions confirm that this is a typical property of solutions to the ASP with i.i.d. distributions and

uniform waiting and idling costs for all jobs. The dome shape is most pronounced when the ratio of idling to waiting cost coefficients is high. When the opposite is true job allowances are more uniform. Solutions do not exhibit this property for cases in which waiting or idling cost coefficients are not uniform for all jobs and/or job duration distributions are not i.i.d.

For brevity, results for all the numerical experiments performed are not presented here. In all cases though, it was found that job allowances increase for all jobs as n increases when $c_\ell = 0$. However, when nonzero overtime costs are included, changes in job allowances with respect to problem size are not necessarily monotonic. Table 3.4 illustrates the effect for the case of zero overtime costs for different problem sizes. In the table $\Delta x/\mu$ is the ratio of the difference between the maximum and minimum job allowance to the mean job duration. It is useful as a representative measure of the non-uniformity of the job allowances. The results show significant increases in job allowances as n increases when waiting cost coefficients are low compared to idling cost coefficients, but relatively small changes when the opposite is true. As idling cost coefficients increase, $\Delta x/\mu$ decreases, indicating that uniform job allowances (equally spaced appointments) are near optimal in such cases. Also, the results indicate that the change in job allowance for a particular job, as n increases, is increasing at a decreasing rate.

	$n=3$			$n=5$			$n=7$		
(c^s, c^w)	(9,1)	(5,5)	(1,9)	(9,1)	(5,5)	(1,9)	(9,1)	(5,5)	(1,9)
x_1	0.262	1.094	1.809	0.311	1.160	1.808	0.335	1.168	1.808
x_2	0.674	1.203	1.811	0.849	1.336	1.818	0.882	1.349	1.818
x_3				0.881	1.313	1.818	0.985	1.358	1.818
x_4				0.769	1.210	1.809	0.955	1.345	1.823
x_5							0.914	1.310	1.818
x_6							0.784	1.219	1.812
$\Delta x/\mu$	0.412	0.109	.002	.570	.176	.01	.650	.190	.015

Table 3.4: Comparison of job allowances for different problem sizes for 3,5, and 7 jobs with $U(0,2)$.

Figure 3.3 shows the dependence of the aggregation bounds, and the statistical upper

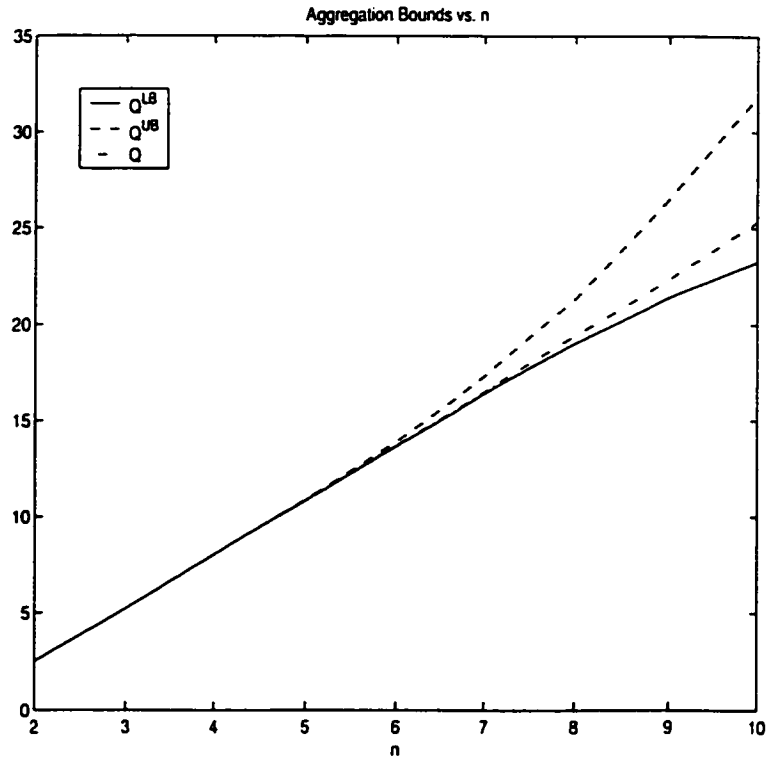

Figure 3.3: Dependence of upper and lower bounds on problem size for $c_i^w = 5$, $c_i^s = 5$, $\forall i$, $c_\ell = 0$, and $U(0, 2)$ job durations.

bound, on problem size. The results indicate that the actual performance, based on the statistical estimate, is typically much better than the worst case bound as problem size grows. Furthermore, the cost is increasing approximately linearly with problem size in the range of problem sizes considered here. Numerical experiments indicate that this linear dependence is not sensitive to relative changes in waiting and idling cost coefficients. Also, from proposition 6, the slope is proportional to the standard deviation of the job durations for the case of i.i.d. job durations.

The results in Table 3.5 illustrate the importance of solving the ASP for different values of the cost coefficients. For each case, $\bar{Q}(\mathbf{x})$ and $\bar{Q}(\mu)$ are estimated by sampling. Together,

(c^w, c^s)	(9,1)	(8,2)	(7,3)	(6,4)	(5,5)	(4,6)	(3,7)	(2,8)	(1,9)
$Q(\mathbf{x})^{LB}$	10.602	17.251	22.097	25.369	27.129	27.496	25.970	22.215	15.030
$Q(\mathbf{x})^{UB}$	11.540	19.324	24.804	28.453	30.661	30.892	29.340	25.553	18.365
$\bar{Q}(\mathbf{x})$	10.736	17.516	22.484	25.816	27.842	27.993	26.577	22.800	15.499
$\bar{Q}(\mu)$	54.509	49.321	45.041	40.888	36.959	32.031	27.727	23.363	18.953
Rel. VSS	407.72%	181.58%	100.32%	58.38%	37.74%	14.42%	4.32%	2.46%	22.28%

Table 3.5: results for 7 jobs with $N(5, 1)$ job durations.

the estimates are used to approximate the *value of the stochastic solution* (VSS), the difference between the optimal solution and the mean-value solution, which can be interpreted as a measure of the importance of solving the ASP. In the table, the *relative* VSS is reported (the difference between the LSB algorithm solution and the mean-value solution, shown as a percentage of the LSB solution). It is clear that relative VSS is high when waiting cost coefficients are high, or similar to idling cost coefficients. Intuitively this can be attributed to the upward bias in customer waiting, i.e., waiting for customers arriving early in the sequence tends to increase waiting of later customers. As waiting time cost coefficients decrease with respect to idle time cost coefficients, the VSS is initially decreasing, and then increasing again. In fact the mean-value solution performs relatively well under some cost structures. From proposition 6 it follows that $\bar{Q}(\mathbf{x})$, $\bar{Q}(\mu)$, and VSS are linearly increasing in the standard deviation of the job durations.

	$\Gamma(6,18)$			$U(1.775, 4.225)$			$N(3, 0.5)$		
(c^w, c^s)	(9,1)	(5,5)	(1,9)	(9,1)	(5,5)	(1,9)	(9,1)	(5,5)	(1,9)
x_1	3.990	3.120	2.285	3.992	3.196	2.155	3.952	3.167	2.268
x_2	4.056	3.352	2.784	4.002	3.412	2.818	3.995	3.369	2.818
x_3	4.055	3.340	2.830	4.002	3.382	2.853	3.989	3.348	2.844
x_4	3.994	3.199	2.726	3.991	3.259	2.718	3.954	3.227	2.752
$Q(\mathbf{x})^{LB}$	5.577	13.059	6.341	4.421	13.286	6.736	5.056	12.734	6.657
$Q(\mathbf{x})^{UB}$	5.779	13.225	6.422	4.422	13.323	6.779	5.073	12.830	6.736

Table 3.6: Comparison of distributions with $\mu = 3$ and $\sigma^2 = 0.5$.

Many heuristic rules suggested in the literature utilize only the mean and variance of the job durations. Table 3.6 contrasts different distribution types and shows that optimal schedules may depend on higher moments. The results are for i.i.d. gamma, uniform, and normally distributed job durations, with mean 3.0 and variance 0.5 for each distribution. The uniform and normal distributions are symmetric, whereas the gamma distribution is skewed (the skewness of $\Gamma(6, 18)$ is 0.55). Comparison of the solutions for the different distribution types indicates that dependence of $Q(\mathbf{x})$ on the distribution type is most pronounced when waiting cost coefficients are high relative to idle cost coefficients. For instance, the relative difference between $\Gamma(6, 18)$ and $U(1.775, 4.225)$ for $(c^w, c^s) = (9, 1)$ is approximately 25%. However, the actual solutions, $\mathbf{x}$, are virtually independent of the distribution type. For the other cost structures in Table 3.6, $(c^w, c^s) = (5, 5)$ and $(1, 9)$, the solutions vary somewhat with respect to changes in the distribution type but the relative changes in $Q(\mathbf{x})$ are typically less than 5%.

3.5.3 Allocating Block Time for Deferrable Surgeries

In this example we illustrate the benefits of solving the ASP in the context of optimizing the allocation of block times for sequences of deferrable surgeries. We assume that there are five blocks scheduled at a given OR in an 8 hour day, and that the session lengths are i.i.d. with distribution $\Gamma(1.0, 1.5)$ (gamma distributed with shape parameter $\lambda = 1.0$, and scale parameter $a = 1.5$). A common heuristic used by OR managers is to allocate block times such that the total time allocated is equal to the mean of the sum of the surgery durations in the session, and to reserve time at the end of the day to avoid overtime costs. For example, setting block sizes equal to the mean in this example results in 1.5 hours for each block and a total of 7.5 of 8 available hours. Thus the surgical team, patients and other resources for the first scheduled block would be coordinated to arrive at the beginning of the day. The start time for the second block would be scheduled 1.5 hours later, and

subsequent blocks scheduled in a similar fashion. Typically there are no direct costs for OR idling (cost of OR up-time is a sunk cost), rather the goal is to trade off the number of surgeries scheduled, costs of idling surgical teams and material resources, and overtime costs. In this example we assume that an equal weight is assigned, i.e., $c_i^w = 1$, $\forall i$, and $c_\ell = 1$. The stopping criteria for the LSB algorithm was set at a tolerance of .01. Under this cost structure the optimal block sizes are

$$x_1^* = 1.563,\ x_2^* = 2.065,\ x_3^* = 2.022,\ x_4^* = 1.703$$

and $Q(\mathbf{x}^*) = 4.14$. Conversely, the mean-value approach ($\bar{x}_i = 1.5$, $\forall i$) yields $\bar{Q}(\mu) = 4.84$. Thus there is approximately a 14.5% reduction in total overtime and surgical team idling associated with using an optimal schedule. The improvement can be viewed as allowing an increase in the effective capacity of the OR. For example, assume that each session has three scheduled surgeries distributed as $\Gamma(1.0, 0.5)$ (thus the sum of durations is distributed as $\Gamma(1.0, 1.5)$). Increasing the number of surgeries in the last session by 1/3 corresponds to a session duration distributed as $\Gamma(1.0, 2.0)$. In this case solving the ASP yields the following block sizes:

$$x_1^* = 1.550,\ x_2^* = 2.051,\ x_3^* = 2.013,\ x_4^* = 1.701$$

and $Q(\mathbf{x}^*) = 4.65$. The result is an increase of about 6.7% in the effective capacity of the OR while still maintaining a reduction in cost compared to the heuristic approach. On the other hand, increasing capacity using the heuristic approach yields a cost of $Q(\mathbf{x}^*) = 5.372$. Due to the high costs of delivering surgical care at hospitals discussed in section 1, such improvements can represent significant savings.

3.5.4 Evaluation of Heuristics for Large n

Another important application of appointment scheduling is in the coordination of the arrival times of patients at outpatient clinics. For such problems n is likely to be much larger.

Tables 3.7 and 3.8 give numerical results that illustrate the performance of the approximate methods discussed in section 3.4. Both tables are for the case of $n = 19$; Table 3.7 assumes normally distributed job durations and Table 3.8 assumes gamma distributed job durations. Given the dependence on the variance of job durations, established in proposition 6, the results represent a rather comprehensive study of the i.i.d. case. Comparisons are made to statistical estimates of the recourse function at the solution obtained from the heuristics. Since $Q(\mathbf{x}^{SEP}) \geq Q(\mathbf{x}^*)$ and $Q(\mathbf{x}^{SJB}) \geq Q(\mathbf{x}^*)$ these estimates are approximate upper bounds. They also contrast the solutions obtained using the SJB heuristic with that from separating the $n = 19$ problem into three $n = 7$ job problems. As shown in section 3.4, both approximations yield lower bounds on the optimum (the lower bound on the latter is the sum of the lower bounds for the three $n = 7$ job problems). Thus the lower bounds, together with the statistical estimates, bound the accuracy loss due to the approximations. For comparison, statistical estimates of the mean-value solution are provided as well.

(c^w, c^s)	(9,1)	(8,2)	(7,3)	(6,4)	(5,5)	(4,6)	(3,7)	(2,8)	(1,9)
$\bar{Q}(\mathbf{x}^{SJB})$	32.394	54.604	72.142	85.503	94.376	99.953	95.964	82.168	57.108
SJB LB	31.590	50.393	62.585	69.542	71.810	69.542	62.585	50.393	31.590
$\bar{Q}(\mathbf{x}^{SEP})$	32.283	53.688	69.959	81.825	89.790	97.085	99.087	94.349	74.354
SEP LB	31.805	51.753	66.291	76.107	81.387	82.488	77.910	66.645	45.091
$\bar{Q}(\mu)$	299.376	272.649	241.756	208.879	177.053	149.621	119.440	88.550	58.653

Table 3.7: Performance of SJB and SEP heuristics for $n = 19$ and i.i.d. $N(5,1)$ job durations.

In the tables, SJB LB and SEP LB are the deterministic, i.e., not sampling based, lower bounds obtained from using the heuristics. It is clear from comparison of the statistical estimates that the heuristics considerably outperform the mean-value solution when waiting cost coefficients are high, or similar to idling cost coefficients. In this case both heuristics

(c^w, c^s)	(9,1)	(8,2)	(7,3)	(6,4)	(5,5)	(4.6)	(3,7)	(2,8)	(1.9)
$\bar{Q}(\mathbf{x}^{SJB})$	61.189	100.078	130.462	154.959	167.813	167.548	152.143	121.938	78.170
SJB LB	55.696	80.808	93.419	97.429	94.654	86.100	72.342	53.642	29.911
$\bar{Q}(\mathbf{x}^{SEP})$	59.896	93.825	116.825	134.879	146.353	155.561	157.188	146.940	108.596
SEP LB	58.272	89.457	109.266	120.639	124.824	121.719	110.568	90.306	57.207
$\bar{Q}(\mu)$	419.900	379.833	333.716	291.121	248.956	207.470	165.769	124.341	81.675

Table 3.8: Performance of SJB and SEP heuristics for $n = 19$ and i.i.d. $\Gamma(1,2)$ job durations.

provide rather tight lower bounds on the optimal solution. In such cases the actual performance of the two approximations is similar, although, the SEP heuristic is better. On the other hand, when idling cost coefficients, are high relative to waiting cost coefficients the mean-value solution performs almost as well as the SJB heuristic, and both outperform the SEP approximation. In all cases SEP LB is a tighter lower bound than SJB LB, although the SJB heuristic outperforms the SEP heuristic when idling costs are high relative to waiting costs.

From proposition 6 the solutions in Tables 3.7 and 3.8 are linearly increasing in a for transformations of the form $\mathbf{Z} \mapsto a\mathbf{Z} + \mathbf{b}$. The *relative* differences between the objective functions at the solutions are independent of these transformations. Thus the above results represent a comprehensive examination of the i.i.d. job duration case. A *worst-case* average measure of the performance of the heuristics can be obtained using max{SJB LB, SEP LB} (the best lower bound) to approximate the optimum. This yields an upper bound on the average gap between the heuristic solutions and the optimal solutions of 16.83% for the SJB heuristic and 24.42% for the SEP heuristic.

3.6 Summary and Conclusions

In this Chapter we provided a new formulation of an important and common problem. The stochastic linear programming model allows considerable flexibility in modeling different types of cost situation. For example, the formulation can easily be generalized to accommodate piecewise linear cost structures, not only for overtime costs, but also for waiting and idling costs. It is also easily extended to include application to systems with special characteristics, such as customer tardiness or "no shows", through adjustments to the performance metrics and the modeling of job duration distributions.

For problems with $n \leq 10$, solutions with tight upper and lower bounds can be obtained with little computational effort. As the problem size grows, the gap between the bounds increases. However, the solutions are typically much better than the worst case bound. For solving large problems, two heuristics have been proposed and tested. Both provide tight lower bounds on the optimum, and both significantly outperform the mean-value solution for cases in which waiting costs are high, or comparable to idling costs. From a practitioner's point of view, the SJB heuristic is amenable to spreadsheet-based implementation, and outperforms (in many cases significantly) the mean-value solution, for all cost structures and job duration distributions considered.

Numerical experiments indicate that VSS is high when either the relative cost of overtime/idling is very high, or when the relative cost of idling is high. There are, however, ranges of cost parameters where choosing job allowances equal to mean job durations is a reasonable approach. That happens when the unit cost of waiting is about 10 to 50% of the unit cost of idling. In general, there are significant cost advantages associated with using optimal appointment schedules. Results for small problems indicate that optimal solutions, although mostly dependent on mean and variance, exhibit some dependence on other distribution properties as well, such as skewness.

The cost of operating an appointments-based service system rises as job durations become more variable, suggesting that in such a case other forms of managing service delivery (segmenting job population into more homogeneous divisions) might be necessary. There is evidence that optimal sequencing, where possible, delivers cost savings that are decreasing in unit overtime costs. If overtime costs are sufficiently high, all jobs experience positive waiting times with high probabilities, and the sequence has negligible impact on makespan.

From a modeling perspective, a potentially valuable extension would be to include probabilistic service level constraints. For instance, these could be in the form of constraints on the probability of makespan exceeding the allocated time for a set of jobs, or similar constraints on individual waiting and idling times. Another important extension would be to model scheduling of start times for activities in a project network. Such a model would be useful for scheduling resources (e.g. construction crews) when the start times of jobs are contingent on the completion times of multiple predecessors, or for coordinating the release of raw materials into a manufacturing system.

Chapter 4

Inventory Placement in the Steel Industry

4.1 Introduction

More than 100 million tons of steel are produced annually in North America with an estimated value of over 50 billion dollars [1]. Steel is an essential raw material for buildings, automobiles, household appliances, and a wide range of consumer and producer products. The Steel industry is widely considered vital to global economic competitiveness and national security It is also considered a mature industry, and often the quintessential example of the *old economy.* Yet there have been significant changes in production technology in recent years that have lowered the barrier to market entry and intensified competition. For example, *mini-mills* use newer electric arc furnace (EAF) technology to process scrap steel. A typical mini-mill consists of a scrap storage area, EAF, and a continuous casting machine. Mini-mills produce between three hundred thousand and one million tons of steel annually and have capital investments measured in tens of millions of dollars. Integrated steel manufacturers (ISM), on the other hand, carry out all of the processes necessary to convert raw

[1] Statistics in this paragraph were obtained from "Steel Industry Technology Road map", American Iron and Steel Institute, February 1998.

ore into finished products. They have dozens of semi-fabrication processes, typically produce three to four million tons of steel annually, and have several billion dollars in capital investment.

Mini-mills are cost-efficient, but restricted in the variety of steel grades they can produce. Nevertheless, they have generated unprecedented competition in the market for plain carbon steels. In response to this competitive pressure, some ISMs that have the technology to produce exotic grades, and to customize finishing operations, have positioned themselves in the high-end markets for exotic/custom-finished steel products. However, customers in these markets demand not only a unique product, but also deliveries that are synchronized with their own production processes. Thus, ISMs are under pressure to increase the variety of products they produce while at the same time increasing their responsiveness to market demand. Even in those cases where product portfolios have not expanded, their composition is turning over much more rapidly. For example, more than 50% of the items in one steel manufacturer's portfolio, consisting of thousands of unique end products, have been introduced in the last 10 years.

Managing variety has become the key to profitability for many ISMs. Whereas product proliferation is a common problem facing many industries, it poses a particularly difficult challenge for ISMs that have long operated in the make-to-order (MTO) production mode. Their production processes are designed to make steel in high volumes in order to minimize setup costs (a brief description of a typical ISM's production processes can be found in section 4.2). Thus, invariably, order fulfillment times are long, ranging from 10 to 15 weeks. However, markets in which ISMs have greater price latitude demand custom-products as well as shorter and more reliable delivery lead times, in the range of 5 to 6 weeks. These requirements are not consistent with the assumptions of low-variety and high-volume production upon which ISMs production processes were built. As a result, where management intervention has been slow, increased product variety has resulted in capacity shortages, as

well as an exploding inventory of semi-finished and finished goods. This has increased operating costs and worsened delivery performance for some customers. A potential solution is capital expenditure to increase the agility of the production system. However, technological and financial constraints make this an unpopular option.

ISM managers see strategic inventory management as a challenge as well as an opportunity to improve operations. Strategically placed inventories of the right semi-finished products, and in the right quantities, can be used to achieve shorter and more reliable delivery times, while still preserving production efficiencies. In effect, this changes the pure MTO architecture into a hybrid make-to-stock (MTS) and MTO architecture, in which a portion of the finished products are made from existing stock of semi-finished products. However, deciding which products to keep in stock, and how to manage their inventories, is far from easy. It is complicated by capacity, yield, and demand uncertainty, process and efficiency related constraints, and the fact that the production processes allow for a continuous range of semi-finished products. There can also be several potential staging areas for the placement placing semi-finished goods inventory. It is clear that the steel industry needs an optimization-based approach to managing product variety.

This Chapter describes a model, and its implementation at one ISM, that helps to choose which semi-finished products to manufacture to stock. It is discussed in the context of a single inventory staging point, but, it can be extended to cover multiple stages. The model accounts for production efficiency requirements, inventory storage policies, and the need for short order-fulfillment lead times encountered at a typical ISM. It is motivated by discussions with senior planners in several different ISM functional areas. Participants have included, for example, inventory managers, purchasers, production planners, caster schedulers, and capacity managers. Although the model is clearly inspired by application to one particular ISM, it is generalizable for application to other ISMs, and to other industries with similar process architecture, for example, the pulp and paper industry.

The Chapter is organized as follows. In the next section we provide some background on the steel-making process, some common process and policy constraints, and factors affecting the design and control of a slab inventory system. In section 3. we formulate and discuss properties of a model for choosing the specific design of slabs to manufacture to stock, given capacity constraints, and uncertainty in customer demand. In section 4 we discuss implementation issues and heuristics, and provide numerical examples based on empirical data obtained from the particular ISM on which we have focused. In section 5 we summarize the managerial implications of our findings and discuss future research directions.

4.2 Background

As opposed to discrete parts manufacturing, in which a manufacturer might utilize many components and sub-assemblies to produce a few finished products, steel making is a *few-to-many* industry. It uses a few raw materials to produce a large variety of finished products. Product differentiation increases as raw material proceeds on its journey toward finished product form. At each production stage, there exist process and efficiency related constraints, as well as stage-specific sources of uncertainty. Naturally, inventory is routinely maintained to act as a buffer between various production stages, to improve efficiency, and to satisfy process constraints. However, only the finished goods inventory is directly tied to improving delivery performance. Our goal is to explore options for staging strategic inventory at earlier points in the production process for the purpose of increasing responsiveness to customer demand. But first, we begin with a general description of how steel is made.

4.2.1 The Steel-Making Process

ISMs produce a variety of finished products, most commonly in the form of *flat rolled steel coils*, or *band* for short. Production of these products is achieved through two basic stages: primary production and finishing. Primary production refers to the conversion of

raw materials (e.g. iron ore, coke and limestone) into band. Finishing operations, on the other hand, make surface and structural modifications to the band to achieve customer specifications on an order. Finishing operations might include surface modifications like galvanization, tin plating, chromium-coating, and painting, as well as shaping operations such as tube-forming.

A typical ISM has a plant with the following primary production operations: coke ovens, blast furnace, vacuum degas stations, a continuous caster, and a hot strip mill. Figure 4.1 illustrates the flows between these operations. The first step in steel production is *iron-making*. This process involves the separation of iron from iron ore. It is carried out through a series of exothermic chemical reactions in a blast furnace. Next, the liquid iron, together with additional scrap steel and various catalysts and purifying fluxes, is reduced in an oxygen furnace and subsequently transferred to a *ladle*. The ladle is then transferred to Ladle Metallurgy and Vacuum Degassing. At this stage various alloying elements may be added to the ladle to modify the chemistry, purification processes are carried out, and additional processing is done to ensure a homogeneous chemistry throughout the ladle.

A batch of liquid steel, called a *heat*, typically varies in size between 100 and 300 tons. The grade of steel is based on its chemical composition and determines the physical properties of the eventual finished product. For example, grade often determines the ductility, tensile strength, and surface quality of the product. Specifying the grade is the first step in customizing the finished product. After the chemistry requirements have been met, the next step is *casting*. This facilitates the transformation of steel from liquid to solid stage. A *continuous caster* has a *tundish* into which liquid steel is poured from the ladle. The liquid steel flows down through an opening in the tundish into one or more water cooled *moulds*. As the steel moves through the mould it is cooled and forms a solid shell. Typically the dimensions of the mould can be adjusted. In the production of slabs (rectangular cross section) it is common to have a fixed thickness but adjustable width. The mould is set to a

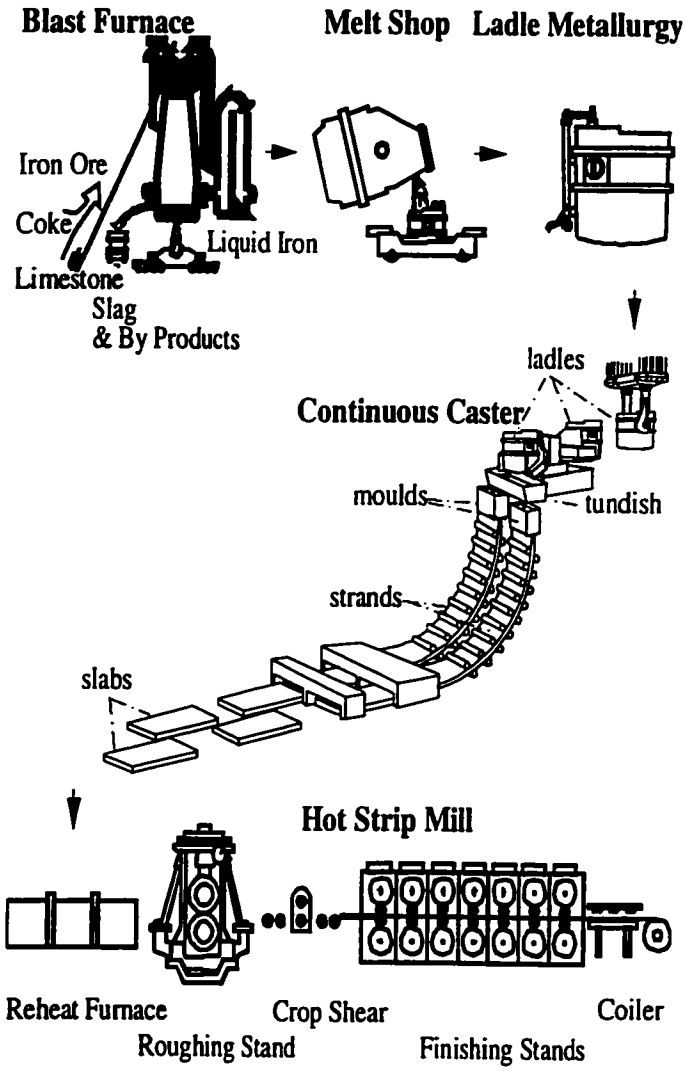

Figure 4.1: Schematic representation of primary operations.

specific width and after the slab exits the mould it is cut with torches to a desired length. Given the slab width the length is the parameter that controls the weight of the slab. The width and weight of the slab are chosen based on the order dimensions of the finished coil specified by the customer.

Within each batch/heat a range of quality levels of slabs are produced. This occurs because the level of the liquid steel in the tundish decreases between ladle changeovers. As a result, some impurities from the surface (slag) mix with the liquid steel that flows through the moulds, resulting in decreased quality of steel. Some slabs are therefore not suitable for customers that have more stringent quality requirements.

Theoretically, slabs can be cast in any width and cut to any length. However, in practice, large rapid width changes are expensive because they result in oblong shaped slabs which have limited applicability to customer orders. Also, ISMs prefer to cast wide rather than narrow widths, since wider slabs have higher throughput at the caster. Similarly, within order specifications, they prefer to cut slabs as close to the specified weight as possible since higher weights require cropping downstream, while lower weights can represent a revenue loss and/or an increased number of pieces to be handled and processed at the hot-mill. In summary, the controllable attributes of a slab up to this point are the grade, width, weight, and internal and surface quality. It is not feasible to rework and correct deficiencies if any of these attributes are out of range with respect a customer order, or the allowable hot-mill tolerances.

After casting is complete, finished slabs may either be labeled and sent to a storage yard for later use or taken directly to the hot-mill for processing. The industry term for the latter practice is *hot-connect*. Heat retained by recently cast slabs reduces the energy and time required for re-heating the slab. Re-heating is necessary to bring the slabs to the right temperature for hot-rolling. Depending on the type of slab and volume of demand for it, ISMs may try to schedule slab production to make hot-connect possible, particularly

for wider slabs since wide slabs take longer to heat. The degree to which hot-connect is practiced depends on the extent to which re-heat furnace is a bottleneck in the material flow.

The hot-mill is a flow line in which a slab is first heated in a furnace to the desired temperature. Next, it is moved on a conveyor to a system of rollers which are used to draw it out into a strip. At this point some width reduction may be carried out by *roughing*, a process by which pressure is applied to the edges of the slab while it is rolled. The amount of width reduction that is possible depends on the metallurgical, process, and end-product quality constraints. The steel strip is subsequently spun into a coil, labeled, and sent to storage for cooling.

Any of a number of finishing operations may be performed on the coil. For example many applications call for electro-plating or painting. Typically, before these operations are carried out the coil goes through a process of *pickling*, in which the surface is exposed to an acid to remove surface imperfections and improve adherence of a surface coating. In many cases *annealing* and/or *tempering* processes, in which the slab is heated and cooled at a controlled rate, are necessary to adjust physical properties. In some cases significant structural modifications may be called for as well; for example, hydro-forming technology is used to produce tube-form products used in the automotive industry.

From this description of the steel-making process, it is clear that there are broadly three categories of inventories. These are in the form of slabs, hot-band, and finished items. Slabs are the least differentiated products; finished goods are fully differentiated. A typical ISM supplies thousands of unique finished products in response to tens of thousands of unique customer orders each year. Similarly, within slab and hot-band categories, there are ranges of potential specifications, called *designs*, which may be used to fill each order. It is therefore difficult to decide which of these are "good" candidates for MTS production mode. In this Chapter, we describe an approach that can be used to answer such a question. Also, we

discuss the implementation at one ISM where it is being used to choose slab and hot-band designs. In order to understand what is a good design, we begin by explaining the relevant sources of uncertainty in the steel making process.

4.2.2 Sources of Uncertainty

Being the suppliers of an important raw material for many industries, steel manufacturers experience a great deal of demand uncertainty. This is caused, in part, by the fact that ISMs supply steel to customers in many different industries, and their demand is affected by all of the factors that affect their customers' demands. However, even for industries in which end-user demand is virtually constant, the existence of long supply chains creates the empirically observable *bullwhip effect* (see, e.g., Lee, Padmanabhan and Wang, 1997, and Chen, 2000). This effect refers to the increased variance of demand between customers and supplier when moving upstream in the supply chain. One reason for this effect is due to batch ordering and order timing. For instance, although end customer demand for a product may be nearly constant, an intermediate supplier may find it desirable to place orders to upstream suppliers in discrete batches at well spaced intervals in time. Steel manufacturers face the brunt of this variability by virtue of their position at near the beginning of many supply chains. Order batching by downstream players in the supply chains in which ISMs participate is commonly believed to be the most significant reason why their orders are much more variable than the demand for end-products.

In some cases relationships with suppliers can make accurate forecasts possible through closer sharing of information (e.g. demand data, inventory levels, production processes). However, this is the exception rather than the norm. Typical lead time for the production of a finished product from raw materials is in the 10-15 weeks range. Long lead times make forecasting less reliable and intensify the bullwhip effect. They also increase the possibility of order cancellations and/or changes. Even though slabs are produced after orders are

received, orders are *confirmed* on much shorter notice (typically 5 weeks prior to the delivery date). It is not uncommon for customers to cancel orders or change order sizes and/or specifications at the time of confirming the order. This is well after slabs corresponding to the original order specifications have been produced. Thus, long production lead times are a source of significant uncertainty in demand.

In addition to being long, replenishment times are also highly variable. This variability is caused, in large part, by the presence of random yield. Because of the large number of operations at an ISM there are many points in the production process at which a yield loss can occur. For example, slabs produced could deviate from the desired grade, and such quality glitches are determined only after the slabs have been cast. After a yield loss occurs it is often too late to make up for it since other products are already scheduled on the caster, or on the hot-mill in the case of hot-band. On account of high setup costs, it is not economical to adjust caster/hot-mill sequences at short notice. Thus it can sometimes take several weeks before the additional production can be scheduled to recover from a yield loss.

4.2.3 Slab Inventory and Storage

Semi-finished inventory is carried within supply chains to buffer the disparity in supply and demand between production units. At a typical ISM there are two stages at which it may be beneficial to carry semi-finished inventory: slab and hot-band. In this section, and throughout the remainder of the Chapter, we concentrate specifically on issues surrounding the slab stage. Similar analysis carries over to the hot-band stage as well.

Slab inventory is carried as a buffer between two major production units at an ISM, the continuous caster and the hot mill. It is clear from the previous Subsection that some slab inventory is unavoidable and some is necessary for production efficiency. Unavoidable inventory may be generated in the form of low grades that are produced between ladle changeovers, and the mid width slabs that have to be cast in going from wide to narrow for

which there may be no existing demand at the time of production. Furthermore, inventory may accumulate in some cases due to changes in customer orders that are made after production of the slabs. Marketing departments are encouraged to find customers for such slabs and planners are encouraged to apply such slabs to orders as soon as possible.

Production efficiency-related reasons that make *planned* slab inventory important are as follows.

- Whenever there is a grade change at the caster, there is a mixture of two chemistries created which makes a portion of the cast steel unsuitable for most applications. Slab inventory increases efficiency by allowing larger batches. It minimizes grade changes and the concomitant loss of material. At one ISM, it is policy to produce minimum three-heat lots of the mid to high demand volume slabs to reduce scrap.

- On occasions when the caster is scheduled to produce narrow slabs, ISMs produce a batch in excess of existing demand to reduce the expense of future setups.

- Slab inventory is necessary to bridge the near uniform production rate at the caster with the batch production mode at the hot-mill.

- Inventory of slabs promotes hot-connect by providing insurance against yield loss. Recall that in order to make hot-connect possible, caster and hot-mill schedules have to be synchronized and any shortfall in supply of slabs can delay the order by several weeks.

Once slabs are in stock, planners are encouraged to apply them to released orders whenever possible. Slab inventories have increased in step with increasing product differentiation.

In addition to providing efficiency gains and a buffer against capacity variations, there are also important strategic reasons for making some slab designs to stock. The production of slabs accounts for roughly half of the time required to process an order. Thus, having

the slabs available when an order arrives can reduce delivery lead times significantly. Lead time benefits can also result from the carrying of finished goods inventory. However, slab inventory is much cheaper to carry because less value has been added at that stage, and because it has a much lower rate of spoilage due to surface corrosion such as rust. Also, there is much greater flexibility in matching slabs to orders than at later stages of production. Thus, significant risk pooling benefits can be realized by carrying slab inventory. Order matching flexibility comes from a combination of process capabilities and flexible customer order specifications. This is explained in detail in the next Subsection. after we discuss the types of storage used to stock slabs.

There are two common ways of storing slabs. In the *random access* area slabs of different designs (grade, width, weight, internal and/or external quality) are stacked in the same pile in a random order. The piles are adjoining, and the heights are restricted because of the need for structural stability. Here it is possible that slabs with similar or identical dimensions may be stored at different locations. Random access storage is necessary for low volume slabs. These are produced in small quantities, but literally thousands of designs are carried. Tracking and retrieval are often difficult in the random access area. In contrast, in the second type of storage, slabs of identical dimensions are stacked in piles in the slab storage yard and are referred to as *clone banks*. Several piles are stacked next to each other, and because the piles contain identical slabs their heights are kept uniform by rotating the picking of slabs. Thus structural stability is maintained and the maximum heights of the piles can be much greater (often as much as 5 times the height of piles in random access storage). The second method therefore benefits from much higher density storage, simpler control and tracking, and shorter retrieval time. However, only high volume slabs are economical to store in this manner. Most ISMs allocate a relatively small proportion of total storage space to clone banks.

4.2.4 Order Matching Flexibility and Cold Application Rules

Slabs are required for each order that arrives at an ISM. Customer orders specify the dimensions of finished hot band that could be applied to fill the order. The hot band dimensions, in turn, translate into a set of slab designs that could be matched to the order. For the most part there is no flexibility in the grade and quality of the slab that is used. However, there are ranges of width and weight for the slab that can be applied to the order. There are two primary reasons why that is the case. First, ISMs have the ability to reduce width at the hot-mill via roughing (see section 4.2.1). Thus slabs that are greater than the specified width for an order may be applied. However, there are limits on the amount of width reduction possible which depend on things such as the grade, width, cast duration, and gauge of the coil required for the order. Second, customers accept coils that fall within a range of weights. Typically aim weight is specified but it is permissible to deviate from this somewhat. Most customers permit weights which are lower than the specified weight but not higher due to constraints at customer loading docks; thus having a higher weight requires the scheduling of an additional cropping operation downstream. Put differently, each customer order translates into a range of slab widths and weights that could be used to match to the order. We call this *order-matching* flexibility.

In some cases, it may be possible to increase order-matching flexibility by downward substitution, i.e., by using slabs of higher internal and external quality than the ones specified by the customer order. However, by and large, downward substitution (known in the steel industry as *grade and quality consolidation*) is not practiced. This is because customers can discern the quality of steel supplied to them and once they receive a higher quality product than what they paid for they demand the same combination of quality and price in future transactions. In short, slab width and weight are essentially the only two attributes of order-matching flexibility.

For an order that is released and is not part of a hot-connect sequence, a planner determines whether there is an existing (cold) slab in inventory that could be used to fill it. This is done by checking if any of the slabs in inventory satisfy a set of rules, called the *cold-application rules*, for the order. If there is such a slab it is then assigned to the order; otherwise a custom-fitted slab design is included in the future production schedule. Similarly, in the event of a yield loss within a hot-connect sequence the affected order(s) is re-released, and an appropriate cold slab is pulled from inventory if one is available.

4.3 The Approach

We propose a two-step approach to the management of product variety at ISMs. The approach is suitable for ISMs that have experienced an increase in the size of their product portfolios, as well as pressure from their customers to improve delivery performance (both delivery time and reliability). Typically, such ISMs carry finished goods inventories, either as a contractual obligation to their customers, or as safety stock to cover mismatches between supply and demand. The first step of our approach consists of moving most of this inventory to earlier production stages where inventory is cheaper to hold, and where benefits of inventory pooling can be exploited to reduce safety stock requirements. Carrying strategic inventory of slabs, for example, is expected to cut the delivery lead time of strategically important customers to less than half of where it stands now. This is highly desirable for customer retention. In fact, it is quite possible that some customers will be willing to pay a premium for having the option to specify order quantities no more than 5-6 weeks in advance of delivery.

The second step of our approach is to develop a plan for the management of inventory of selected semi-processed items that will be made-to-stock. We plan to replenish stock by scheduling regular production/purchase of certain items in each planning period. This

will free the remaining production capacity to be used to satisfy orders in a *reactive* mode. Overall, the approach is expected to simultaneously reduce total inventory, improve delivery performance, and improve capacity utilization.

Only the first of these two steps has been completed and implemented, and that is what is described in this Chapter. The inventory management problem is addressed in Chapter 5. We are concerned here with determining which slab designs are "good" candidates for MTS production. For this purpose, we recognize two important functions of slab inventory: facilitating hot-connect and reducing delivery lead times. In either case, we are interested in identifying designs that could cover the maximum number of customer orders and thus minimize inventory carrying costs. Most, if not all, of these designs are stored as clone banks due to the high volume of the requirements for them. Extensions of that work reported here might include the development of an appropriate model-based inventory management system that would include multiple inventory staging points for determining how the stock of MTS slabs and other types of semi-finished inventory should be chosen and replenished. Key issues here are: limited production capacity at the caster, seasonal demand pattern, and planned and unplanned downtime, e.g., due to scheduled maintenance and repair of furnace lining.

4.4 Model Formulation

In order to maintain low inventory a small number of slab designs are chosen for MTS production. In the model discussed below we treat the following two concurrent objectives for carrying planned slab inventory:

- Slabs are to be carried for the purpose of reducing lead times and improving on-time-delivery performance.
- Slabs are to be carried as insurance against yield losses at the caster or hot-mill.

The former allows efficient expediting of back-logged orders and shorter quoted lead times; the latter helps on-time delivery of orders that are subject to hot-connect by allowing production planners to pull cold slabs in the event of a yield loss.

The description of the model that we have developed is as follows. Let p denote the maximum number of designs that are to be made to stock. The number p is determined by available storage space. Later, we show the results of a sensitivity analysis to determine the incremental benefit of making p larger. J denotes the set of potential slab designs, and $C = \{1, ..., p\}$ the index set of chosen slab designs with widths, w_j, weights, u_j, grades, g_j, and qualities, q_j. We define $I = \{1, ..., m\}$ to be the set of all orders within a historical data set. For each slab $j \in J$, and order $i \in I$, we assume there is a nonnegative reward r_{ij}, which is strictly positive if the cold-application rules are satisfied for the slab-order pair and zero otherwise. Given a slab and order that satisfy the cold-application rules, the size of the reward may depend on, for example, variable production cost, order size (in tons), importance of the customer, and width and weight discrepancy between the slab and the ideal width-weight combination for the order. Given the r_{ij}, our problem is to choose the index set C of cardinality p that maximizes total reward, i.e.,

$$\max_{C \subseteq J} \{ \sum_{i=1}^{m} \max_{j \in C} \{r_{ij}\} \mid |C| \leq p \}. \tag{4.1}$$

Problem (4.1) is complicated by the fact that there is a continuous range of potential slab widths and weights, i.e., the set J is continuous. As long as the width-weight combination of a slab lies within the range specified by the cold-application rules, it is a feasible design. Thus, for each order there is an infinite number of feasible slab designs.

For an order indexed i, let (w_i^{min}, w_i^{max}) and (u_i^{min}, u_i^{max}) denote the pairs of minimum and maximum feasible widths and weights, respectively. Our first simplification of the problem is to demonstrate that under the following assumption, which is quite reasonable from a practical viewpoint, the set of potential designs that need to be considered is finite.

We begin by temporarily redefining the rewards in the following way. Let $r_i(w,u)$ denote the reward associated with applying a slab of the required grade and quality with width and weight (w,u) to order i. Furthermore, we make the following assumption:

Assumption 1 *The rewards $r_i(w,u)$ are linear with respect to slab width and weight (w,u), for each $i \in I$, where $w_i^{min} \leq w \leq w_i^{max}$ and $u_i^{min} \leq u \leq u_i^{max}$.*

For the particular application in mind this assumption is valid since rewards are reasonably assumed to be roughly linear in width and weight for the following reasons

- Cast duration is linearly proportional to width and thus yield is linearly increasing in width.
- Revenue is linearly proportional to slab weight.

Given this assumption the problem of optimizing over the set of potential slabs for a given subset of orders, p, can be written as

$$\max_{w,u}\{\sum_{i\in p}^{c} r_i(w,u) \mid w_i^{min} \leq w \leq w_i^{max},\ m_i^{min} \leq u \leq m_i^{max},\ i \in p\}. \tag{4.2}$$

Since there is at least one common design (4.2) is feasible and assumption (1) guarantees that a solution such that the width and weight correspond to a corner point solution. In practice, this means that J can be reduced to a finite set of potential designs. This result demonstrates an important property of the problem. It implies that we can start with a finite set of slab designs, and search within that set, to find the best p designs. It parallels the *node optimality* property of certain location problems (see for example Chapter 2, pp. 75-78, of Mirchandani and Francis, 1990). In a location theory context, node optimality refers to the fact that, under reasonable assumptions, if one needs to locate at most p facilities to maximize profits from serving n geographically dispersed customers ($n \geq p$), then optimal locations are chosen from the set of demand nodes.

The fact that each subset of designs can be associated with a single optimal solution to (4.2) implies a finite set consisting of p optimal designs. We demonstrate the logic behind this through some specific examples. Consider first a single order from the set I, say order 1. In this case, the set of feasible designs can be shown as a rectangle encompassing the ideal width-weight combination for the order. The latter is shown as an open circle in Figure 4.2. The reward $r_1(w, u)$ is linear over all feasible (w, u), and the unique optimal design for order 1 is denoted by a star in Figure 4.2. In this example the set $I = \{1\}$ is a singleton. The claim that there is a finite set of potential slab designs holds true in this case since we can choose the set S_1 to have a single member, the optimal design.

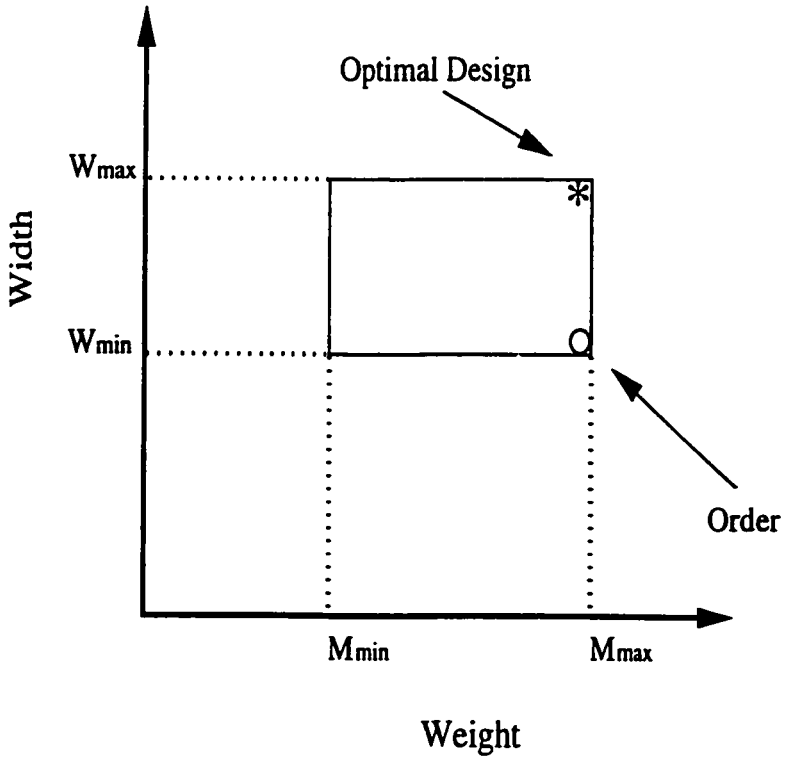

Figure 4.2: Order-matching flexibility.

Notice that in the example illustrated in Figure 4.2, the ideal width-weight combination is shown at the upper right corner of the feasible region of slab widths and weights. This is

based on actual considerations at one ISM but it no doubt has considerable generality. The ISMs clients will accept a somewhat lower weight, but never higher because of limitations at their loading docks. The time and material cost of scheduling the necessary downstream cropping operation for overweight coils is prohibitive. It is always desirable to avoid this non-value adding step. Thus the ordered weight is assumed to be the maximum possible one, and it is this weight that becomes the goal in order to simultaneously maximize revenues and minimize the number of pieces handled/processed downstream. The location of the optimal slab width depends largely on which production process is currently the bottleneck. For example, when caster capacity is a bottleneck, ISMs try to cast slabs with the highest possible width to maximize the total tonnage that can be processed through the caster. Thus, Figure 4.2 shows the ideal width-weight combination in one instance of the slab design problem. This may change over time, or from one application to another. Our model accommodates such changes so long as assumption 1 is satisfied.

Next, consider an order subset consisting of five orders, as shown in Figure 4.3. We immediately notice that there is no single slab design that can satisfy all five orders. Therefore, the feasible set is empty and trivially finite. If we consider the two subsets consisting of two and three orders shown in Figure 4.3, we notice that each of these subsets can be satisfied via common slab designs. For illustrative purposes, this figure assumes that the caster is indeed the bottleneck, and therefore maximum width and weight slab designs (top right corners) are preferred. Then, the set of designs that contains the complete set of optimal designs is the set of top right corners of all slab design sets that are feasible for at least one member of the original subset. In the three order example, this set includes (a) optimal designs for each particular order (b) optimal designs for combinations of two orders from the original subset, and (c) a single design that is optimal with respect to all three orders from the original subset.

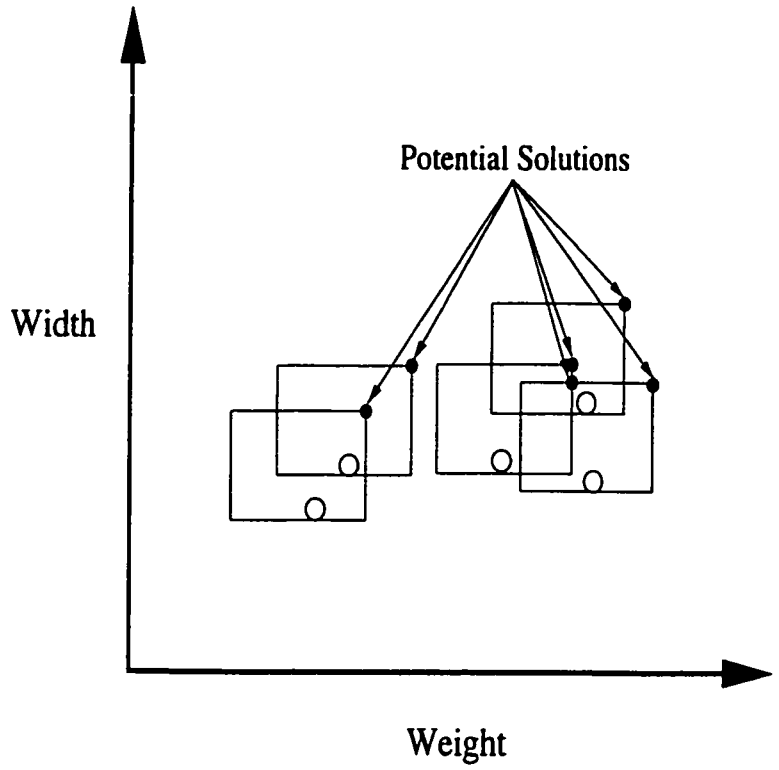

Figure 4.3: Illustration of discrete set of potential solutions when r_{ij} linearly increasing in slab width and weight.

Since the order set is finite, there is a finite number of non-empty order subsets. Therefore, there is a finite set of possible slab designs, which we denote as $J = \{1, ..., n\}$ from this point forward, within which lie the p best designs. Thus, the problem can be formulated as a combinatorial optimization problem in which the relationships between slabs and orders is represented by a *bipartite* graph such as the example in Figure 4.4. Arcs have nonnegative weight, r_{ij}, and arcs with zero weight (ones that do not satisfy the cold- application rules) are ignored.

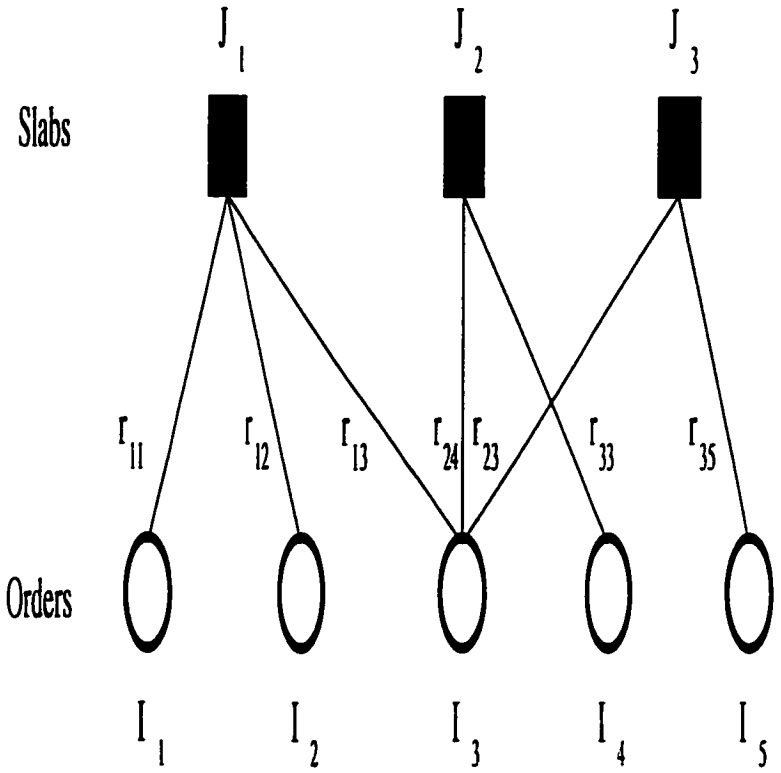

Figure 4.4: Graph representation of the set-covering problem in slab design optimization.

In the language of mathematical optimization, we need to determine values of binary decision variables x_j and y_{ij} in order to maximize the total reward. That is, if

$$x_j = \begin{cases} 1 & \text{if } j \in J \text{ is chosen} \\ 0 & \text{otherwise} \end{cases} \qquad y_{ij} = \begin{cases} 1 & \text{if order } i \text{ is assigned to slab } j \\ 0 & \text{otherwise} \end{cases}$$

and assuming that each order should be assigned to at most one chosen slab design, the problem can be formulated as

$$\max \quad Z = \sum_{i=1}^{m} \sum_{j=1}^{n} r_{ij} y_{ij} \tag{4.3}$$

$$s.t.$$

$$\sum_{j=1}^{n} y_{ij} \leq 1, \quad \forall i \tag{4.4}$$

$$\sum_{j=1}^{n} x_j \leq p \tag{4.5}$$

$$y_{ij} \leq x_j, \quad \forall(i,j) \tag{4.6}$$

$$y_{ij}, x_j \in \{0,1\}, \quad \forall(i,j). \tag{4.7}$$

Up to this point we have not commented on the precise form of the rewards, r_{ij}, associated with applying a slab design to an order. The factors that determine an appropriate way of modeling these depend on the particular application. In the next section we discuss a simple definition which can be quite useful in practical applications. First, we point out the following interesting property of the model. Assume that the rewards are stochastic and can be written as the sum of a deterministic part and a random part, $r_{ij} = \bar{r}_{ij} + \xi_{ij}$, where ξ_{ij} is observed after the choice of designs but before the allocation of orders to designs. For example, they may have an underlying dependence on some random variable(s) (e.g. demand). The structure of the problem is that of a two-stage stochastic linear program where the variables x_j correspond to first stage decisions, and y_{ij} are the second stage (recourse) decisions. The stochastic linear program has the following interesting property:

Proposition 2 *The stochastic linear programming analogue to (4.3) - (4.7) is equivalent to the associated mean-value-problem*[2] *if* $\xi_{ij} = \xi_i$, *i.e.,* ξ_{ij} *depends only on the order* i *and not the design* j.

Proof: Since the second stage decision is

$$y_{ij}^* = \begin{cases} 1 & \text{if } j = \text{argmax}_{j' \in C}\{r_{ij'}\} \\ 0 & \text{otherwise,} \end{cases}$$

it follows that y_{ij}^* is independent of ξ_{ij} and therefore

$$E_{\xi}[\sum_{i=1}^{m}\sum_{j=1}^{n} r_{ij}(\xi_i) y_{ij}^*] = \sum_{i=1}^{m}\sum_{j=1}^{n} E_{\xi}[r_{ij}(\xi_i)] y_{ij}^*$$

□

[2]see Chapter 2, section 2, for definition of the mean-value problem

Thus the introduction of randomness into the reward coefficients in this special case does not change the underlying problem structure, i.e., the mean-value solution is optimal. In the next section we use this property in the application of the model.

Problem (4.3 - 4.7) is well known in other contexts. For example, the same mathematical formulation is obtained when a firm needs to choose at most p locations to maintain bank accounts for customer payments, given fixed costs for opening such accounts. In that context, the problem is called the *lock-box* problem (see, for example, Cornuèjols, Fisher and Nemhauser, 1977, for a discussion). In general, the problem can be described as an instance of the fixed-charge network flow problem. A special case of the problem in which facilities are located at demand nodes (i.e. $J = I$ in the above notation), and the fixed costs for opening facilities are zero, is called the *p-median* problem in location theory (see Mirchandani and Francis, 1990, Chapter 2, for a review). The *p-median* problem is typically posed in the context of determining the location of (at most) p facilities such that profits from serving a range of geographically distinct customers is maximized.

While the problem defined by (4.3) - (4.7) is a well known problem, it is also known to be *NP-Complete* (Cornuèjols, Fisher and Nemhauser, 1977). In practice, that means it is at least as difficult as a set of other problems in combinatorial optimization for which no exact and reasonably fast algorithms are known for solving large instances of the problem exactly. The size of the problem that is relevant for a typical ISM is indeed quite large. For example, a typical historical order set may contain tens of thousands of distinct orders. Similarly, the number of slab designs that we need to evaluate, albeit finite, may easily run into hundreds of thousands, and the size of clone bank inventory, although small, may accommodate fifty different designs. Thus, there is little hope of solving a realistic problem of this kind. We also recognize that ISM managers would like to perform sensitivity analysis by solving different versions of the problem that represent different reward functions (recall that this depends primarily on which process in the production system is a bottleneck, and

bottlenecks change depending on order mix). They also need to continually re-apply this technique to determine whether there have been sufficient changes in the customer order portfolio to warrant a change in the slab designs kept in stock.

Naturally, many operations researchers have made contributions by developing heuristics that work well for particular problem cases. A discussion of solution methods can be found in Nemhauser and Wolsey (1999), pages 495-512. A well known, fast, and easily implemented heuristic is the *greedy dual-ascent heuristic.* It has been studied in detail by Cornuèjols, Fisher and Nemhauser (1977). They provide an analysis of the worst case performance of this algorithm. Letting Z^* denote the optimal solution and Z^G the solution obtained by the greedy heuristic, they prove the following

Proposition 3 (Cornuèjols, Fisher and Nemhauser, 1977)

$$\frac{Z^G}{Z^*} \geq (1 - (\frac{p-1}{p})^p) \geq \frac{e-1}{e} \approx 0.63. \tag{4.8}$$

Cornuèjols, Fisher and Nemhauser (1977) also show that there exists a class of problems for which this bound is sharp, but, that there is empirical evidence that the average performance is $\frac{Z^G}{Z^*} \approx 0.8$. Greedy heuristics such as this are often implemented as the first of a two-step procedure in which an initial solution is generated (construction) in step 1, and subsequent improvements are made in step 2. A well known heuristic for step 2 is an interchange heuristic due to Teitz and Bart (1967). This heuristic is based on a neighborhood search of an existing solution to determine pairwise interchanges that yield improved solutions. The algorithm terminates when no improving interchange can be found.

For problems of large size, a combination of the greedy heuristic to obtain a "good" initial solution, and the interchange heuristic to improve this solution, can be a fast method for obtaining near optimal solutions. There is evidence in the literature that the greedy-interchange heuristics often gives near optimal results. For these reasons, we have chosen

this two-pronged approach. The solution obtained using this approach is, by definition, a lower bound on the optimal total reward. An upper bound on the optimal reward is obtained by using the Lagrangian dual formulation, which is solved using a sub-gradient algorithm (see Appendix A.1). The gap between the upper and lower bounds informs us about the quality of our heuristic solution. While the worst case bound on the greedy heuristic in proposition 3 is not very good, our numerical experimentation reveals that the greedy-interchange heuristic produces a very good solution to the problem. It is also fast and easy to implement, even on an inexpensive personal computer. The next section describes results of numerical experiments and discusses implementation issues.

4.5 Implementation and Numerical Examples

The model developed in section 4.4 has been applied at a particular ISM and it is currently being used as the basis of a decision support tool for choosing which slab designs should be stored as clone-bank inventory and for making decisions concerning external purchasing of slabs. Initial implementation and testing were carried out on a Sun Ultra 10 Workstation with 128 MG of Ram using C++. The greedy-interchange heuristic solution was subsequently transferred to a Windows NT platform and a suitable graphical interface was added to accommodate various data management needs and to provide a suitable level of abstraction for non-technical users.

A diagram illustrating the different types of data required by the model is provided in Figure 4.5. The model uses various sources of data at the ISM to formulate the optimization problem proposed in the previous section. An instance of the model is defined by the set of rewards for applying orders to slabs. The order data is used to generate a list of distinct *order types*, each of which has distinct specifications. To generate instances of the model consistent with the needs of the decision makers the following assumptions were made

- For each distinct subset of orders that could be covered by a single common slab design the maximum slab width and slab weight were considered. This is consistent with the assumption that Continuous Caster is the bottleneck in the production process (see Figure 4.3 for an illustration).

- All the distinct order types over a specified planning horizon (e.g. 3-6 months) were rolled into a single time period. Robustness of chosen designs is later tested by studying the proportion of orders (in tons) that can be met with these designs when order data from different periods is used.

- Rewards were assumed equal to estimates of the mean demand for each order type, D_i, over the specified planning horizon, i.e., , $r_{ij} = E[D_i]$, where $E[D_i]$ is an estimate of the mean demand for order type i.

Estimation of the mean demand for each distinct order type can be done using historical order data, quantitative and/or subjective forecasts of demand, or a mixture of the two. The objective is to maximize total demand covered by p designs, given the secondary objective of maximizing caster throughput. The basic structure of the algorithm used is as follows:

i. A finite set of non-redundant potential clone bank designs is generated based on cold-application rules in place at the ISM and an appropriate matrix of rewards for the particular problem instance is generated and stored using a sparse matrix storage scheme.

ii. The greedy-interchange heuristic is applied to generate a user defined number of clone bank designs.

iii. Supplemental data analysis is carried out and a range of output is generated for analysis of various properties of the proposed bank (e.g. factors impacting hot-mill scheduling, product and customer breakdown, etc.) and for modeling subsequent

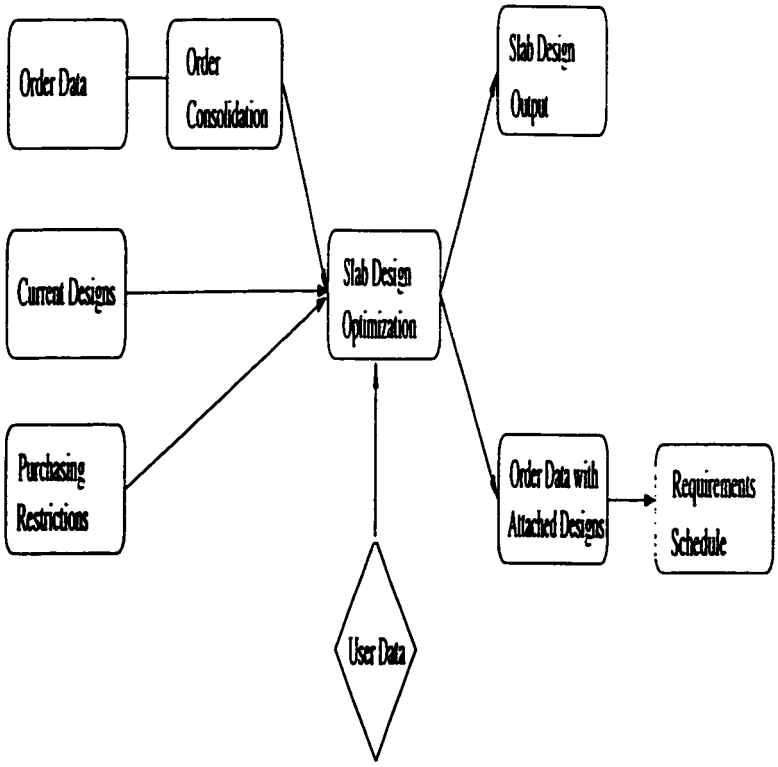

Figure 4.5: Data flow for the slab design optimization model.

replenishment of the proposed bank (mean and variance of demand). This data can subsequently be used for feedback into step i.

Numerical analysis was carried out using historical data from a particular ISM. For confidentiality reasons no specific data regarding order trends or chosen slab dimensions is given. The results are intended only to give some idea of the types of analysis that can be done using the above model.

The results are presented in Table 4.1. They illustrate the typical accuracy of the two heuristics employed. Comparison with the Lagrangian dual upper bound (last column in the table) indicates that the greedy + interchange heuristic typically finds a solution that is within 2% of optimal. This is significantly better than the worst case bound in

proposition 3. Intuitively this can be attributed to two factors. First, the cold-application rules restrict the applicability of slab designs to orders, and thus the choice of a given slab design has a more localized impact on other design choices than in general location applications. Second, although there is a broad range of potential designs, there appears to be a natural aggregation (similar to the well known Pareto rule) of a large proportion of orders into a small set of width/weight ranges. As a result the greedy-interchange heuristic provides a fast and satisfactory solution method for the structure and size of problems considered here.

p	Greedy Heuristic	Greedy_{UB}	%Gap	Greedy + Interchange Heuristic	Subgradient Decomposition	% Gap
10	22.751	33.130	31.3	23.342 (12)	23.473	0.56
20	33.130	49.175	32.6	33.144 (6)	33.699	1.65
30	40.964	59.764	31.5	40.979 (6)	41.581	1.54
40	46.562	63.885	27.3	46.685 (8)	47.522	1.76
50	50.756	67.384	24.7	50.770 (10)	50.793	0.04
60	54.340	71.739	24.2	54.902 (21)	55.769	1.55
70	57.425	76.210	24.6	57.988 (21)	58.577	1.00
80	60.059	77.820	22.8	60.620 (30)	61.143	0.85
90	62.439	81.104	23.0	62.974 (27)	63.488	0.81
100	64.628	84.215	23.2	65.198 (30)	65.736	0.82
150	73.450	92.580	20.7	73.794 (34)	74.608	1.09
200	79.707	98.211	18.8	79.833 (15)	81.158	1.63

Table 4.1: Numerical results for several values of p for greedy and greedy + interchange heuristics, and the solution of the Lagrangian dual (upper bound).

The running time of the greedy heuristic is dependent on the form of the cold-application rules that determine the extent of applicability of slabs to orders. Examples of computation time for various problem sizes are given in Table 4.2. Typical running time for the greedy heuristic for problems in Table 4.1 was less than 15 minutes on a Sun Ultra 10 and comparable results can be achieved on a typical PC (366 MHz, 128 MG Ram). The interchange heuristic was found to yield small improvements to the greedy solution ($< 3\%$); however

it significantly increased computation times. The number in brackets in column 3 of Table 4.1 denotes the number of iterations after which the interchange heuristic stopped, i.e., no interchange could improve the solution any further. Many possibilities exist for choosing the search criteria for the interchange heuristic. In this case we used single interchanges where the entering slab design was the first to improve the solution and the exiting design was the one that yielded maximum improvement.

Algorithm/p	10	20	30	40	50	60
Greedy	60	80	121	156	190	223
Interchange	924	2558	4771	5917	9013	13548
Subgrad. Decomp.	7818	8569	8586	8189	9253	8617
Subgrad. Decomp. Iterations	1976	1993	2000	2000	2000	2000

Algorithm/p	70	80	90	100	150	200
Greedy	265	310	345	399	559	762
Interchange	14018	18270	21508	25978	99496	75346
Subgrad. Decomp.	9083	8888	9201	9460	8528	9418
Subgrad. Decomp. Iterations	2000	1916	2000	2000	2000	2000

Table 4.2: Sample computation times in seconds for several values of p for greedy and greedy + interchange heuristics, and the solution of the Lagrangian dual (upper bound).

After applying the algorithm to one instance of the problem, i.e., one set of 6-month order data, the robustness of the solution was tested by calculating the percentage of orders (in tons) that could be covered by the best 50 designs in different planning periods. It was found that the choice of designs is quite robust, and that the percent of orders covered is approximately constant across different planning periods. We expect this not to hold for long periods of time, due to portfolio turnover. Therefore, the software application of the slab design optimization problem has the functionality to start with some initial designs and add the best $p - x$ design to the already existing x designs. Such flexibility accommodates the potential need for incremental changes to the portfolio over time.

Figure 4.6 illustrates the diminishing return for increasing the number of clone bank

positions. It was found, using the model, that more that 50% of orders in 1999 could have been filled using 60 different designs. However, as the number of available cells is increased, a long tail develops. Managing this tail is a problem confronted by all ISMs. It illustrates the fact that there are typically many orders that require custom slab designs. This representation is consistent with the 80-20 rule or Pareto distribution of many naturally observable phenomena, including, for example, the ABC classification of inventory (see, for example, Silver, Pyke and Peterson, 1998).

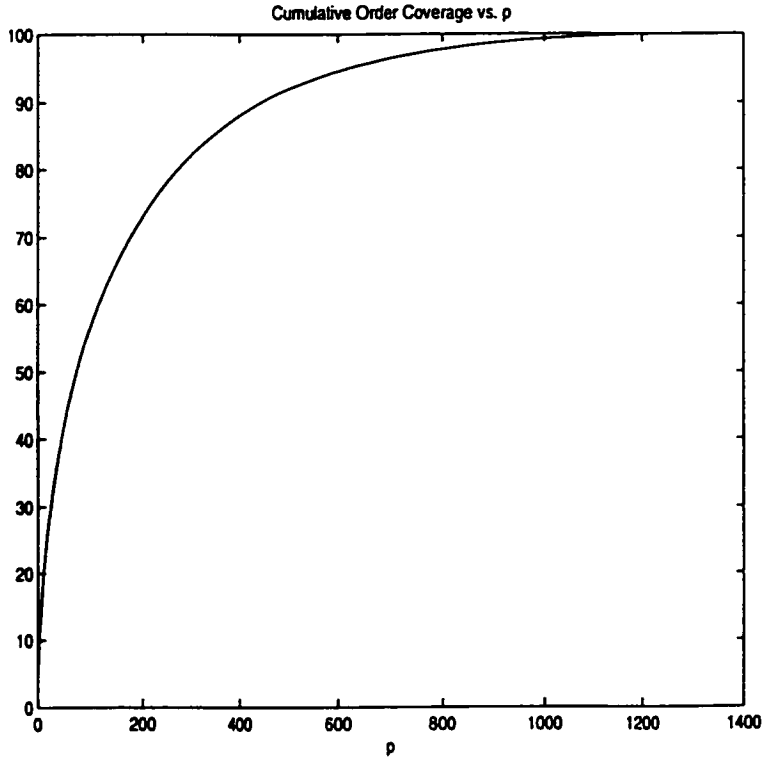

Figure 4.6: Cumulative percentage coverage of total demand for slab designs obtained using the greedy heuristic.

Figure 4.7 is an example of a typical requirements schedule for a particular slab design over a 6 month period. The schedule was evaluated using the set of orders assigned to the particular design by the greedy heuristic. The order due dates were then adjusted

according to production-routing-dependent processing times to determine the approximate week in which slabs for each order would have been required for processing at the hot-mill. The figure is typical of requirements schedules for slab designs in the steel industry. It can be observed that there is some base load, which depends on the season, mixed with a high variance component of demand. Such schedules can also be used to approximate the probability distribution of the weekly demand for a particular slab design, and subsequently used to analyze the effectiveness of different types of inventory control policies.

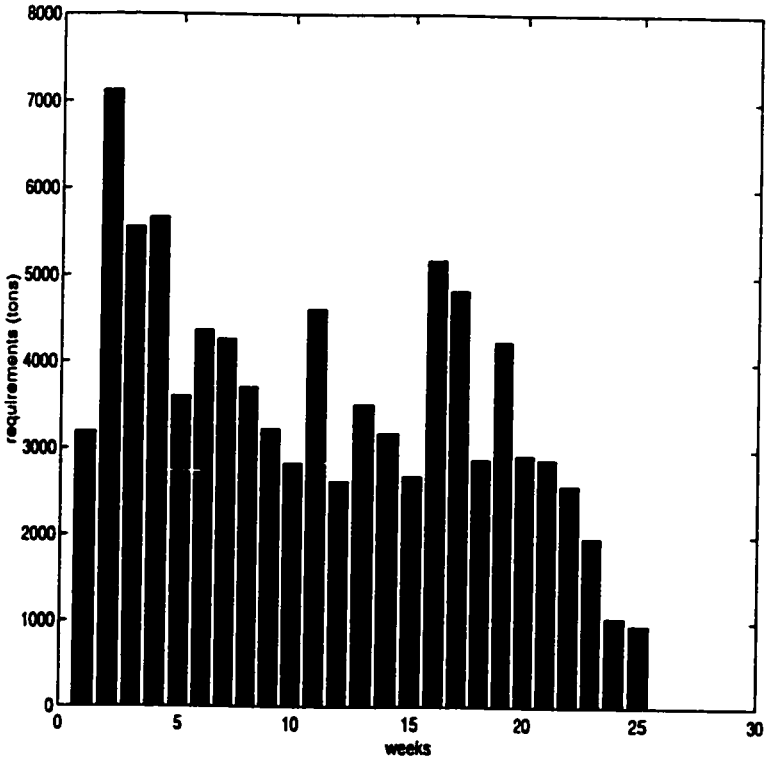

Figure 4.7: Example of a typical requirements schedule for a particular slab design over 25 weeks in 1999.

4.6 Summary and Conclusions

We have motivated and provided evidence of the implementability of a well known combinatorial optimization model for making inventory placement decisions in a continuous process industry. Due to the recent need for ISMs to reduce order-fulfillment lead times, such a model can play an important role as a tool for planning slab inventory. Numerical examples indicate that allocating even very limited space for planned inventory may allow for significant improvements in order-fulfillment lead time. Also, we have demonstrated that near optimal solutions can be expected even for the very large-scale problems considered here, using heuristics that are fast and easy to implement.

The model presented in this Chapter represents a *first-cut* at modeling the choice of clone bank designs. It ignores capacity limitations at individual cells, i.e., how many tons of slabs each cell can hold, as well as decisions and policies regarding how clone bank inventory is to be replenished in the presence of uncertain future demand and yield. These factors may affect the choice of a set of clone bank designs. In the next Chapter we discuss a more detailed stochastic optimization model for optimizing the inventory level of chosen slab designs.

Chapter 5

Inventory Deployment Under Uncertainty

5.1 Introduction

Advance planning of inventory requirements is a difficult and common problem faced by managers of manufacturing systems operating under a *make-to-stock* (MTS) policy. Inventory planning decisions depend on the penalties for excesses and shortages of inventory with respect to random demand and product yield, as well as replenishment lead time, and the frequency and duration of setups. Typically for low and medium demand volume products the inventory policy is a tradeoff between fixed setup/transaction costs and inventory holding and shortage costs. High demand volume items, on the other hand, are produced in each planning period, and therefore the frequency of setups is largely fixed, obviating the need to explicitly consider fixed costs. The complexity of the problem is significantly increased when planning decisions must be coordinated across multiple items for reasons such as common capacity constraints, budget restrictions, and substitutable demand.

In the steel industry the customized nature of finished products, and the continuous range of designs of semi-finished inventory, expand the planning to include the choice of

which designs to stock. Chapter 4 considers a specific problem in the steel industry in which semi-finished inventory designs are chosen for the purpose of shifting from make-to-order (MTO) to a hybrid MTO/MTS production mode. When the number of design choices is limited by a storage space constraint the problem is equivalent to the well known *p-median* problem. The model in Chapter 4 is *uncapacitated*, i.e., it implicitly assumes perfect matching of supply and demand. In reality, however, the presence of long production lead time requires inventory levels to be chosen long before demand materializes. Thus costs due to supply/demand mismatch are unavoidable.

The goal of this chapter is to study the impact of uncertainty in yield and demand on inventory deployment in the steel industry. The model we propose is motivated by the application described in chapter 4. However, it is applicable to other process industries and to other problem contexts. In the general framework we refer to the different types of stock-keeping-units (slabs in the example in Chapter 4) as *designs*, and the requirements for them as *demand*. We use the terms *customer-class* and *order-type* interchangeably to refer to specifications of an order that affect inventory matching. Rules for allocating designs to order types (cold-application rules in Chapter 4) are referred to simply as application rules. Acquisition of inventory, whether by in-house production or external purchase, is called *procurement*. The decisions about which designs to stock, which orders to support, and inventory levels, are referred to separately as *design-choice*, *order-choice*, and *lot-size* decisions, respectively, and together as *inventory deployment*.

The model we propose in this chapter is formulated as a two-stage stochastic linear program with binary first stage decision variables representing both the choice of designs and the choice of order-types to be supported through the make-to-stock (MTS) production mode. First stage decisions also include the quantity of inventory to plan. Each of these decisions are made prior to the resolution of uncertainty in yield and demand. Once the uncertainty is resolved, available supply is allocated to demand to maximize profits subject

to the application rules governing allowable allocation of designs to order-types.

The main contributions of this chapter are as follows. We show that a common relaxation of the stochastic programming model corresponds to the p-median problem, and therefore, is NP-complete. Structural properties of the deterministic equivalent problem and the second-stage recourse problem are presented that permit significant reduction in the number of discrete decision variables and lay the groundwork for efficient heuristics. Applicability of a greedy allocation algorithm to the recourse problem is discussed, and a stylized example illustrating the potential worst case error is presented. Due to the large size of problems found in industry, exact algorithms are not practical. Instead we propose two heuristics that exploit the structure of the model, and which are applicable to large-scale instances of the problem. A series of numerical experiments are presented which establish the accuracy of the heuristics. Results for an instance of the problem from the steel industry (based on actual data) are used to motivate the economic importance of the model, and to illustrate some important managerial insights.

The Chapter is organized as follows. In the next section we provide a brief review of relevant literature. In section 5.3 we present the model formulation, discuss special cases, and provide insights based on the structure of the problem. We propose and discuss some heuristics in section 5.4 and report, in section 5.5, a computational study of their accuracy; we also provide examples based on empirical data to illustrate the economic importance of the model. We summarize the work presented in the Chapter, and discuss potential extensions of it in section 5.6.

5.2 Selected Literature Review

The model of interest in this Chapter is related to several different areas of literature including: random yield production models, multi-product substitutable inventory models,

and stochastic fixed-charge network problems. We provide a brief overview of this literature in the present section, and draw analogies between our problem and previously studied problems. A majority of stochastic inventory model literature deals with single period (newsvendor) problems, and with two product instances of the substitutable products problem. The literature on stochastic versions of the fixed-charge network problem is very limited. To the author's knowledge there are no articles which incorporate random yield and substitutable product features in the context of the stochastic fixed-charge network problem studied in this Chapter.

Single-period stochastic inventory models assume a two-stage decision process in which an initial inventory level is chosen, random supply and/or demand are observed, and inventory is subsequently allocated to demand. The simplest example is the newsvendor model for which there is a vast literature (see Porteus, 1990, for a review). We make no attempt to present a detailed review of the newsvendor model here, however, some of the extensions include: quantity discounts (Jucker and Rosenblatt, 1985), risk aversion (Eeckhoudt and Schlesinger, 1995), multiple customer classes (Sen and Zhang, 1999), and pricing (Petruzzi and Dada, 1999). The analytical tractability of this model makes it an important building block for more general stochastic inventory models.

The importance of considering uncertain yield is well established. Yano and Lee (1995) give a comprehensive review of factors influencing yield uncertainty and related modeling approaches. In the context of single-product problems there is a considerable literature including Shih (1980), Moinzadeh and Lee (1987), Lee and Yano (1987), and Henig and Gerchak, (1990). The majority of models assume *stochastically proportional yield losses*, i.e., yield uncertainty is represented by an independent random fraction of the lot-size. This typically results in convex stochastic optimization problems, however, it is not appropriate when yield uncertainty is correlated with lot-size. Klein (1966) and White (1967) formulate models as Markov decision processes that explicitly account for this correlation. These

types of models are generally restricted to a small number of products and low dimensional state spaces. Our model assumes stochastically proportional yield losses and is applicable to large-scale problems involving both multiple products and demand classes.

Ignall and Veinott (1969) first considered the multi-product inventory problem with downward (one-way) substitution. They focus on conditions under which a myopic ordering policy is optimal in a multi-period setting. Analytic results for two-product problems given perfect yield, are presented by McGillvray and Silver (1978), Parlar and Goyal (1984). Bassok et al. (1999) consider solution methods for a two-stage stochastic linear programming formulation (2S-SLP) of large-scale multi-product problems with downward substitution. Gerchak and Wang (1996) consider the two-product case in which there is also yield uncertainty. Studies of large scale single-period multi-product substitutable inventory models that account for yield uncertainty have focused on applications to semi-conductor manufacturing. Bitran and Dasu (1992) study heuristics for a lot-sizing model and Hsu and Bassok (1999) present an efficient algorithm for a 2S-SLP model that assumes a single lot-size decision resulting in random yield of multiple products. Our model incorporates all aspects of these substitution models but also generalizes to cases other than downward substitution.

The multi-product inventory problem with substitutable demand is related to the multi-location inventory problem with transshipment between locations studied by Karmarkar (1979), Robinson (1990), and others. A similar problem, the stochastic transportation problem (Williams, 1962), differs from these problems because shipping schedules are fixed prior to realization of random demand rather than after. These problems are related to the general topic of two-stage stochastic linear programming problems with network recourse in which the recourse problem constraints are of the network flow type. Wallace (1986) studied specialized computational procedures for solving these types of problems. Our model is closely related to the work of Wallace since it assumes a two-stage problem with network

recourse representing the allocation of supply to demand. However, it is a generalization because it incorporates binary decision variables in the first-stage decision process that represent selection of supply and demand nodes that define the second stage network flow problem.

There is recent and growing interest in the area of stochastic mixed-integer programming models. See Birge and Louveaux (1997, Chapter 8) and Schultz et al. (1996) for a review of general stochastic mixed-integer programming applications and solution methods, and van der Vlerk (2000) for a recent bibliography. In this context our model can be classified as a stochastic *fixed-charge-network* problem (see Nemhauser and Wolsey, 1999, Chapter II.6 for discussion of deterministic fixed-charge-network problems). Studies of stochastic versions of these models are very limited. Louveaux and Peeters (1992) study a dual based procedure for the stochastic uncapacitated facility location problem. Laporte et al. (1994) study exact solution procedures for a location problem with stochastic demands in which facility capacities (inventory levels) are chosen a priori. Another closely related work is by Rao et al. (2000). They study a multi-product inventory model with downward substitution and fixed setup costs. The differences between their model and ours are that they assume perfect yield, no storage constraints (rather, fixed setup costs), and take advantage of the downward substitution structure to propose simulation based heuristics.

5.3 Model Formulation and Analysis

The model we present is aimed at addressing the following decision process. Given a known (but potentially large) set of inventory designs, a subset is chosen to support the MTS production mode, subject to a storage space constraint. Furthermore, some or all of the orders that are applicable to one or more of the chosen designs may by selected for planning in the MTS production mode, and lot-size decisions are optimized according to these

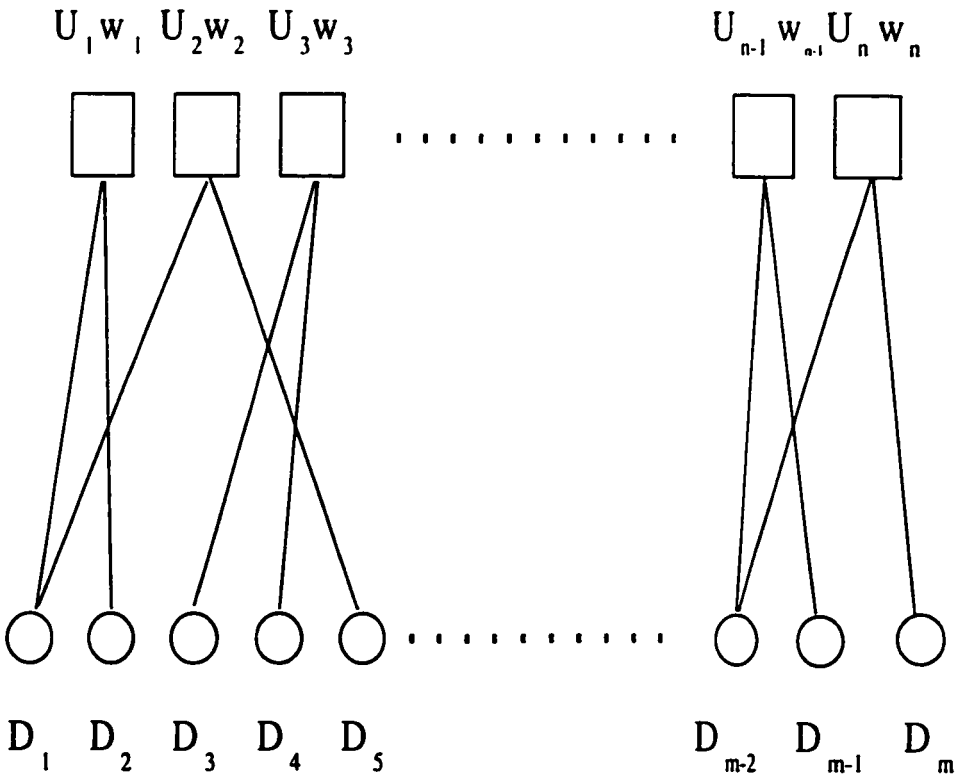

Figure 5.1: Bipartite graph representing the inventory deployment problem

design/order-choices. Any orders not included in the MTS set will be served by the MTO production mode. Also, all orders are eventually satisfied and shortages are fully back-ordered. It is assumed that the design-choice, order-choice, and lot-size decisions are all made before yield and demand uncertainty is resolved. Once these are resolved supply and demand are matched according to a given set of application rules governing allowable allocations of supply to demand.

The structure of the problem centers around a network described by the bipartite graph in Figure 5.1. Vertices in the graph can be partitioned into a set of potential supply nodes, $J = \{1, 2, ..., n\}$, that represent the set of design-choices, and a set of potential demand nodes, $I = \{1, 2, ..., m\}$, that represent the different order-choices. Edges between the supply and demand nodes indicate allowable allocations of supply to demand according to

the application rules. To formulate the model we define the following notation:

c_j^e: per unit cost of having excess inventory of design j.

c_i^s: per unit cost of shortage for order-type i.

r_{ij}: application reward from applying order-type i to design j.

c_j^p: per unit variable production cost for design j.

x_j: binary decision variable representing the decision to stock design j.

q_i: binary decision variable representing whether order-type i is supplied from inventory ($q_i = 1$) or not ($q_i = 0$).

c: number of permitted design-choices

w_j: production/procurement lot-size for design j.

y_{ij}: amount of order-type i supplied by design j.

a_{ij}: incidence parameter that is 1 if there is an edge between i and j and 0 otherwise.

s_i: shortage for order-type i.

e_j: excess product j inventory.

U_j: random yield for design j.

D_i: random demand for order-type i.

$\boldsymbol{\xi}$: random vector with components that are yields, U_j, and random demands, D_i.

where $c \in Z_+$, $c_j^e \in \Re$, $(c_i^s, r_{ij}, c_j^p, w_j, y_{ij}, s_i, e_j) \in \Re_+$, and $(x_j, q_i) \in B$. Note that, as in previous chapters, we use upper case for random variables, and boldface for vectors. Also, Z_+ and B are the set of nonnegative integers and binary values, $\{0, 1\}$, respectively. The random vector $\boldsymbol{\xi}$ has support $\Xi \subseteq \Re^n$ and probability distribution P. It is assumed to have components that are nonnegative and to have finite first moments, denoted by $\bar{\boldsymbol{\xi}}$.

The first-stage decisions are the design and order-choices, $\mathbf{x} \in B^n$, and $\mathbf{q} \in B^m$, and the vector of lot-sizes, $\mathbf{w} \in \Re_+^n$, respectively. In the first-stage there is a total production cost, $\sum_{j=1}^n c_j^p w_j$. In the second-stage there is a cost $\sum_{j=1}^n c_j^e e_j$, for surplus inventory, and a cost

$\sum_{i=1}^{m} c_i^s s_i$, for inventory shortage. The total application reward from matching inventory with demand is $\sum_{j=1}^{n}\sum_{i=1}^{m} r_{ij}y_{ij}$. Thus the complete second-stage objective function is

$$\sum_{j=1}^{n}\sum_{i=1}^{m} r_{ij}y_{ij} - \sum_{j=1}^{n} c_j^e e_j - \sum_{i=1}^{m} c_i^s s_i.$$

For convenience, the dependence of y_{ij}, e_j, and s_i, on the realization of ξ has been suppressed in the notation.

There are second-stage inventory balance constraints on the allocation of inventory to demand of the form

$$\sum_{i=1}^{m} a_{ij}y_{ij} + e_j = U_j w_j, \ \forall j, \tag{5.1}$$

and

$$\sum_{j=1}^{n} a_{ij}y_{ij} + s_i = D_i q_i, \ \forall i. \tag{5.2}$$

The a_{ij} are determined by application rules. The first-stage binary decision variables, q_i, in (5.2), reflect the fact that application rewards and shortage cost penalties are incurred only for designs that are chosen to be covered in the first-stage. (It is implicitly assumed that if an order cannot be covered by a chosen design it is covered by alternative means; for example, as part of the MTO production mode.) The yield dependence in the right hand side of constraint (5.1) implies stochastically proportional yield losses. This is a reasonable assumption for high demand volume designs in the steel industry where production yield losses are approximately linear in lot-size, and yield losses for externally purchased designs are typically of the *all-or-nothing* type. This latter type of yield loss results from mistakes made by external suppliers in the delivery process.

The complete problem involves the discrete selection of supply and demand nodes, as well as lot-sizes for chosen supply nodes, such that total profits are maximized. Putting it all together, and assuming that the firm is *risk neutral*, the complete problem can be expressed as

$$\max\{Z \ = \ -\mathbf{c}^p\mathbf{w} + Q(\mathbf{x}, \mathbf{q}, \mathbf{w})\} \tag{5.3}$$

$$s.t. \sum_{j=1}^{n} x_j \leq c, \tag{5.4}$$

$$\mathbf{x} \in B^n,\ \mathbf{q} \in B^m, \mathbf{w} \geq 0 \tag{5.5}$$

The recourse function is $Q(\mathbf{x}, \mathbf{q}, \mathbf{w}) = E_{\boldsymbol{\xi}}[Q(\mathbf{x}, \mathbf{q}, \mathbf{w}, \boldsymbol{\xi})]$, where $Q(\mathbf{x}, \mathbf{q}, \mathbf{w}, \boldsymbol{\xi})$ is defined by

$$Q(\mathbf{x}, \mathbf{q}, \mathbf{w}, \boldsymbol{\xi}) = \max\{\sum_{j=1}^{n}\sum_{i=1}^{m} r_{ij}y_{ij} - \sum_{j=1}^{n} c_j^e e_j - \sum_{i=1}^{m} c_i^s s_i\} \tag{5.6}$$

$$s.t. \quad \sum_{i=1}^{m} a_{ij}y_{ij} + e_j = U_j w_j,\ \forall j, \tag{5.7}$$

$$\sum_{j=1}^{n} a_{ij}y_{ij} + s_i = D_i q_i,\ \forall i, \tag{5.8}$$

$$y_{ij} \leq D_i x_j,\ \forall (i,j), \tag{5.9}$$

$$y_{ij} \geq 0,\ s_i, e_j \geq 0,\ \forall (i,j). \tag{5.10}$$

We refer to (5.3) - (5.5) as the inventory deployment problem (IDP). It has complete recourse, i.e., it is feasible for any feasible $(\mathbf{x}, \mathbf{w})$, due to the positive linear basis provided by $(\mathbf{e}, \mathbf{s})$ in constraints (5.7) - (5.8), and given that $U_j \geq 0,\ D_i \geq 0, \forall (i,j)$. Furthermore, randomness occurs in the right hand sides only, and second-stage cost coefficients are deterministic. It is assumed that the allocation of inventory is continuous which results in a tractable model, and is also a reasonable assumption, given that the designs chosen typically are those with significant demand volume. In the steel industry there are no significant fixed costs for choosing a design, thus fixed costs for $x_j = 1$ are zero. However, this assumption is trivial to relax for applications in other contexts to which the model is applicable (e.g. facility location problems) where fixed costs may be high. Also, note that there is no explicit constraint forcing optimal lot-size $w_j^* = 0$ if $x_j = 0$. However, this is implied by constraints (5.7) and (5.9) since otherwise $e_j > 0$ and unnecessary additional excess inventory costs are incurred with no additional rewards. Similarly, constraints (5.8) and (5.9) imply $q_i^* = 0$ if all $x_j = 0$ such that $a_{ij} = 1$, i.e., if no applicable inventory design has been chosen.

We make the following assumptions about the objective function coefficients. Shortage costs, c_i^s, are nonnegative, i.e., it is never advantageous to incur a shortage. The excess cost coefficients, c_j^e, may be positive or negative depending on the application. A negative c_j^e would correspond to a positive salvage value for excess inventory. The application rewards are such that if for some (i,j) $a_{ij} = 0$ then $r_{ij} = 0$ as well. Furthermore, it is assumed that $r_{ij} \geq \max\{-c_i^s - c_j^e, 0\}$, $\forall(i,j)$, i.e., it is never advantageous, in the second stage, to choose not to allocate available supply of design j to order i if $r_{ij} > 0$, and, there is some nonzero reward for having a design that can fill an order. It is also assumed that first stage procurement cost coefficients are such that $c_j^p + c_j^e > 0$, since otherwise it is trivially optimal to produce an infinite quantity of design j, and for each design j there is an order-type i such that $r_{ij} > c_i^s + c_j^p$, since otherwise it is optimal not to produce design j at all.

The restriction of the problem obtained by fixing some feasible $\mathbf{x} \in B^n$ is an important problem on its own, which we refer to as the lot-sizing problem (LSP). It corresponds to the case in which there is a known set of designs for which inventory levels must be planned. From a practical standpoint, the IDP pertains to medium and long-range decisions about which designs to stock, whereas the LSP may be solved frequently to control inventory levels. Although the LSP is large-scale in nature, it is computationally less challenging than the IDP because all the decision variables are continuous. It can be solved efficiently using decomposition based approaches (e.g., Bender's decomposition, Dantzig-Wolfe decomposition).

The LSP can be modified to account for initial inventory by adjusting the right-hand side of (5.1), as $U_j w_j \rightarrow w_j^0 + U_j w_j, \forall j$. Since it may be necessary to solve the LSP repeatedly, it is of interest to understand the properties of the optimal policy. Such a policy presupposes a critical inventory level, $\boldsymbol{\alpha}$, such that if $\mathbf{w}^0 \leq \boldsymbol{\alpha}$, then $\mathbf{w}^* = \boldsymbol{\alpha} - \mathbf{w}^0$. The practical benefit, when demand distribution is stationary, is that the LSP needs to be solved only once and $\mathbf{w}^*$ can be trivially obtained for arbitrary $\mathbf{w}^0$. In the absence of uncertainty in yield losses,

i.e., $U_j = \bar{u}_j$ w.p. 1, the optimality condition satisfies the following equivalence relationship (see Kall and Wallace, 1995, chapter 1)

$$\nabla_{\mathbf{w}} Q(\mathbf{x}, \mathbf{q}, \mathbf{w}) = E_{\boldsymbol{\xi}}[\nabla_{\mathbf{w}} Q(\mathbf{x}, \mathbf{q}, \mathbf{w}, \boldsymbol{\xi})] = 0 \Leftrightarrow E_{\boldsymbol{\xi}}[\nabla_{\bar{\mathbf{w}}}[Q(\mathbf{x}, \mathbf{q}, \bar{\mathbf{w}}, \boldsymbol{\xi})] = 0 \tag{5.11}$$

where $\bar{w}_j = w_j^0 + \bar{u}_j w_j$, $\forall j$, and therefore

$$w_j^* = \frac{\alpha_j - w_j^0}{\bar{u}_j}, \forall j. \tag{5.12}$$

Bassok et al. (1999) point out a property analogous to (5.12) for a multi-product stochastic linear programming model with downward substitution and perfect yield. When yields are random, Henig and Gerchak (1990) have proved the existence of a critical value below which an order is placed for the single-product problem. However, they show that in general an order-up-to policy is not optimal, which immediately extends to the multi-product generalization of the problem.

5.3.1 Deterministic Equivalent Problem Analysis

Given the very large-scale nature of problems encountered in practice we assume, from this point forward, that the support, Ξ, is a finite set of scenarios, represented by the random vectors $\boldsymbol{\xi}^k = (u_1^k, u_2^k, ..., u_n^k, d_1^k, d_2^k, ..., d_m^k)$, with associated probabilities p_k, $k = 1, ..., K$. (In section 5 we discuss the reasonableness of this assumption and methods for generating scenarios.) The deterministic equivalent problem can be written as

$$\max\{Z = -\sum_{j=1}^{n} c_j^p w_j + \sum_{k=1}^{K} p^k [\max\{\sum_{j=1}^{n}\sum_{i=1}^{m} r_{ij} y_{ij}^k - \sum_{j=1}^{n} c_j^e e_j^k - \sum_{i=1}^{m} c_i^s s_i^k\}]\} \tag{5.13}$$

$$s.t. \quad \sum_{j=1}^{n} x_j \leq c, \tag{5.14}$$

$$\sum_{i=1}^{m} a_{ij} y_{ij}^k + e_j^k = u_j^k w_j, \ \forall j, \tag{5.15}$$

$$\sum_{j=1}^{n} a_{ij} y_{ij}^k + s_i^k = d_i^k q_i, \ \forall i, \tag{5.16}$$

$$y_{ij}^k \leq d_i^k x_j, \ \forall j, \tag{5.17}$$

$$x_j, q_i \in \{0,1\}, \ \forall (i,j), y_{ij}^k, \ s_i^k, \ e_j^k \geq 0, \ \forall (i,j,k). \tag{5.18}$$

To begin with, we consider a common relaxation of stochastic linear programs, such as the IDP, in which the random right hand sides of the subproblem constraints are replaced by their first moments (Huang et al. 1977). This corresponds to replacing each of the set of subproblem constraints (5.15), (5.16), and (5.17), with the sum of the K rows weighted by their associated probabilities, p^k. This relaxation, known as the mean-value problem (see section 2.2), significantly reduces the size of the deterministic equivalent problem by reducing K subproblems to a single subproblem. The following proposition uses this relaxation to establish the connection between the stochastic model, IDP, and the deterministic p-median problem (chapter 4).

Proposition 1 *The mean-value relaxation of the IDP is equivalent to the p-median problem.*

Proof: Let $\bar{\xi} = (\bar{\mathbf{d}}, \bar{\mathbf{u}})$. Replacing random vector $\boldsymbol{\xi}$ with single realization $\bar{\xi}$, occurring w.p. 1, yields the mean-value-approximation which, for some feasible $\mathbf{x} \in B^n$, has optimal first-stage decisions,

$$q_i^* = \min\{\sum_{j=1}^{n} a_{ij} x_j, 1\} \text{ and } w_j^* = (\frac{1}{\bar{u}_j}) \sum_{i \in C_j} \bar{d}_i x_j,$$

where $C_j = \{i \mid r_{ij} > r_{ij'}, \forall j'\}$. The optimal second-stage solution is $s_i = 0 \ \forall i$, $e_j = 0, \ \forall j$, and

$$y_{ij}^* = \begin{cases} \bar{d}_i & \text{if } i \in C_j \\ 0 & \text{otherwise.} \end{cases}$$

Making the transformation $y_{ij} \rightarrow \bar{d}_i y_{ij}$ transforms the mean-value relaxation of the IDP to

$$\max\{Z_R = -\sum_{j=1}^{n} \frac{c_j^p}{\bar{u}_j} \sum_{i \in C} \bar{d}_i x_j + \sum_{i=1}^{m} \sum_{j=1}^{n} r_{ij} \bar{d}_i y_{ij}\} \tag{5.19}$$

$$s.t. \ \sum_{i=1}^{m} x_j \leq c, \tag{5.20}$$

$$\sum_{i=1}^{m} a_{ij}\bar{d}_i y_{ij} = \sum_{i \in C} \bar{d}_i x_j, \quad \forall j, \tag{5.21}$$

$$\sum_{j=1}^{n} a_{ij} y_{ij} = \min\{\sum_{j=1}^{n} a_{ij} x_j, 1\}, \quad \forall i, \tag{5.22}$$

$$y_{ij} \leq x_j, \quad \forall (i,j), \tag{5.23}$$

$$x_j \in \{0,1\}, \forall j, y_{ij} \geq 0, \quad \forall (i,j). \tag{5.24}$$

Substituting (5.21) into the first term of the objective function yields

$$Z_R = -\sum_{i=1}^{m}\sum_{j=1}^{n} \frac{c_j^p}{\bar{u}_j} a_{ij}\bar{d}_i y_{ij} + \sum_{i=1}^{m}\sum_{j=1}^{n} r_{ij}\bar{d}_i y_{ij} \tag{5.25}$$

$$= \sum_{i=1}^{m}\sum_{j=1}^{n} \bar{r}_{ij} y_{ij} \tag{5.26}$$

$$\tag{5.27}$$

where $\bar{r}_{ij} = r_{ij}\bar{d}_i - c_j^p a_{ij}\bar{d}_i/u_j$. Since $r_{ij} = 0$ if $a_{ij} = 0$ (by assumption) then $\bar{r}_{ij} = 0$ if $a_{ij} = 0$, replacing (5.22) with $\{y_{ij} \mid \sum_{j=1}^{n} a_{ij} y_{ij} \leq 1, \forall i\}$ yields a problem equivalent to the p-median problem except for binary restrictions on the y_{ij}. However, $y_{ij}^* \in B$ follows from the total unimodularity of constraints in the p-median problem, given $\mathbf{x} \in B^n$ (Cornuejols, Fisher, and Nemhauser, 1977). □

Since the p-median problem is a relaxation of the IDP an immediate corollary to proposition 1 is that the IDP is also NP-complete. The p-median problem has been studied in detail (see Mirchandani and Francis, 1990, chapter 2, and references therein). Although reliable exact algorithms for large-scale instances of the problem are not available, fast and easy-to-implement heuristics that yield near optimal results in most cases have been proposed (see chapter 4 for examples of the application of a greedy-interchange heuristic to p-median problem). This relaxation forms the basis for one of the heuristics presented in section 5.4.

Managers of hybrid MTS/MTO production systems in the steel industry must trade off the benefits of reduced lead-times with inventory allocation (shortage and excess) costs

resulting from the assignment of orders to MTS production mode. Orders for which inventory is not planned are assumed to be served by default through the MTO production mode. However, assigning an order-type to the MTS production mode represents a commitment to have available inventory to allocate to that order-type when it arises. In the IDP the decision, q_i, represents whether or not to plan to supply demand for order-type i from inventory. If $q_i = 1$ then, from constraints (5.7) and (5.8) in the IDP, there is the potential for nonzero rewards, r_{ij}, as well as nonzero shortage costs, c_i^s, and excess costs c_j^e. In the following proposition we establish an important property of the discrete nature of these first-stage order-choice decisions which allows considerable reduction in the number of discrete first-stage decision variables and plays an important role in the heuristics discussed in section 5.4.

Proposition 2 *For fixed* $\mathbf{x} \in B^n$ *there is an optimal solution to the relaxation of the IDP, with constraints* $\mathbf{q} \in B^m$ *relaxed, and* $0 \leq \mathbf{q} \leq 1$, *such that* $\mathbf{q}^* \in B^m$.

Proof: We consider the case in which $q_i^* > 0$ (otherwise $q_i^* = 0$) and show that it implies $q_i^* = 1$. Treating the LP relaxation of (5.13) - (5.18) as a parametric program in $\mathbf{q}$ it can be rewritten as

$$Z_{\mathbf{q}}^* = \max\{Q(\mathbf{x}, \mathbf{w}, \mathbf{q}) \mid (\mathbf{x}, \mathbf{w}, \mathbf{y}, \mathbf{s}, \mathbf{e}) \in \mathcal{P}, \sum_{j=1}^{n} a_{ij} y_{ij}^k + s_i^k = d_i^k q_i, \ \forall i, k\} \tag{5.28}$$

where $\mathcal{P}$ = ((5.14), (5.15), (5.17)) has all zero right hand sides and is independent of $\mathbf{q}$. Writing the corresponding dual of (5.28) as

$$Z_{\mathbf{q}}^* = \min\{\pi \mathbf{h} \mid \pi \in \mathcal{D}\} \tag{5.29}$$

where $\mathbf{h} = (q_1 d_1^1, q_2 d_2^1, ..., q_1 d_1^2, q_2 d_2^2, ..., q_1 d_1^K, q_2 d_2^K, ...)$ and $\pi = (\pi_1^1, \pi_2^1, ..., \pi_1^2, \pi_2^2, ..., \pi_1^K, \pi_2^K, ...)$. Since $d_i^k \geq 0, \forall (i,k)$, and $q_i \geq 0, \ \forall i$, it follows that $\mathbf{h} \geq 0$. Thus if $q_i^0 > 0$

$$(\frac{\partial Z_q^*}{\partial q_i})_{q_i = q_i^0} = \sum_{k=1}^{K} \pi_i^k d_i^k > 0$$

is independent of q_i and since (5.29) is unconstrained in q_i other than $0 \leq q_i \leq 1$ it follows that $q_i^* = 1$. □

Thus by proposition 2 the number of binary decision variables can be reduced from $m + n$ to n. This effectively increases the size of instances of IDP that can be solved reliably by exact methods. As we show in section 5.4 it also important in the development of heuristics.

5.3.2 Recourse Problem Analysis

The heuristics proposed in the next section rely on decomposition of the scenario subproblems. It is therefore important to establish the structural properties of the second-stage recourse problem. In the second-stage, for a particular realization of ξ, the inventory levels, yield, and demand are known with certainty, and the problem is to allocate inventory to demand in such a way that total second-stage profit is maximized. Each design is a source node, each order-type is a demand node, and the second-stage problem has network structure corresponding to the *transportation problem*. Special algorithms, such as the primal dual or flow augmentation algorithms, that have polynomial running time are available to solve such problems (Nemhauser and Wolsey, 1999, chapter I.3).

In some special cases the transportation problem can be solved trivially using a greedy-type algorithm. This relies on identification of a sequence of arcs $\{(i_t, j_t),\ t = 1, ..., m \times n\}$ such that the following algorithm yields the optimal solution:

Greedy Allocation Algorithm:

Step 0: $w_j(0) = w^0 + u_j w_j,\ \ d_i(0) = d_i$.

Step 1: *For* $t = 1, ...n \times m$ *do:*

If $a_{i_t j_t} = 1$ *then* $y_{i_t j_t} = \min\{w_{j_t}(t), d_{i_t}(t)\}$ *and* $w_{j_t}(t+1) = w_{j_t}(t) - y_{i_t j_t}$,

$d_{i_t}(t+1) = d_{i_t}(t) - y_{i_t j_t}$

Step 2: *return to Step 1.*

The basic idea of the algorithm is that arcs are ordered such that flows across each are sequentially maximized subject to maximum available supply and demand. Existence of such a sequence can be proved based on the fact that every basis for a transportation problem is triangular (see Corollary 13.2, pp. 382, Murty, 1983). However, the identification of such a sequence is nontrivial. A special condition for identifying an optimal sequence, referred to as a Monge sequence, for the case in which total supply equals total demand was proved by Hoffman (1963). The following proposition is an adaptation of Hoffman's proposition specific to the recourse problem for the special case of no shortages or excesses.

Proposition 3 *(Hoffman, 1963) If the coefficients satisfy the condition that for every* $1 \leq i$, $s \leq m$, $1 \leq j$, $t \leq n$, *if* (i,j) *precedes both* (i,t) *and* (j,s) *in the sequence, then the inequality* $r_{ij} + r_{st} \geq r_{it} + r_{sj}$ *is satisfied, then the sequence is a Monge sequence.*

In the case of the LSP the transportation problem can be viewed as consisting of *dummy* supply and demand nodes for shortages, **s**, and excesses, **e**. Hsu and Bassok (1999) identified a Monge sequence for the special case of downward substitution when r_{ij}, $\forall(i,j)$ are separable into costs depending on the product i and the order-type j, i.e., $r_{ij} = \hat{r}_i^a + \hat{r}_j^a$. There are two reasons why these assumptions do not apply to inventory deployment problems in the steel industry

i. Allocation of designs to orders is not restricted to downward substitution.

ii. The r_{ij} depend on the relative difference between the width and weight of the slab design, and the ideal dimensions of the order-type to which it is applied.

The first follows from greater generality of application rules (see section 4.4 of Chapter

4). The second follows from (a) dependence of revenues on weight discrepancy between an order and design, and (b) the cost dependence of cropping and side-reduction operations on width discrepancy. The following example illustrates the potential error due to greedy allocation when the sequence is not a Monge sequence.

Example: In this example we consider a heuristic in which supply is allocated to demand by choosing arcs in decreasing order of rewards. We assume $n = m = 2$, $r_{ij} > 0$, $c_i^s = 0$, $i = 1, 2$, $c_j^e = 0$, $j = 1, 2$, and $u_j = 1$, $j = 1, 2$. Demands, d_1, d_2, and supplies, w_1, w_2, are known in the second stage, and rewards satisfy $r_{11} = r_{22} = r$ and $r_{12} > r$. The structure of the problem is illustrated in Figure 5.2.

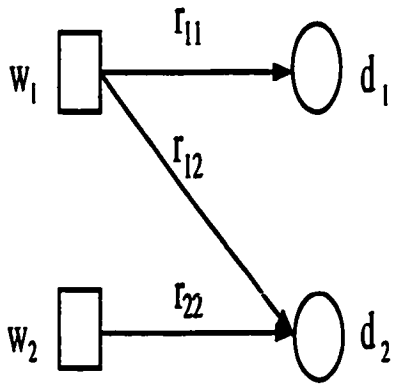

Figure 5.2: Two design/order example

It is straightforward to show that the optimal allocation is achieved by either of the sequences $\{11, 22, 12\}$ or $\{22, 11, 12\}$ if $r_{12} < 2r$. Thus, according to the greedy algorithm

$$Z^* = r \min\{d_1, w_1\} + r \min\{d_2, w_2\} + r_{12} \min\{d_2 - \min\{d_2, w_2\}, w_1 - \min\{d_1, w_1\}\}.$$

Alternatively, for sequences $\{12, 11, 22\}$ or $\{12, 22, 11\}$, the arc with maximum marginal return, r_{12}, is chosen first, and the solution is

$$Z = r \min\{d_1, w_1 - \min\{d_2, w_1\}\} + r \min\{d_2, w_2 - \min\{d_2, w_1\}\} + r_{12} \min\{d_2, w_1\}.$$

The maximum relative difference is achieved when the total allocation across arc 12 could have otherwise been allocated across both arcs 11 and 22. We assume that supply is allocated across arcs in order of decreasing rewards, thus the maximum relative difference between the two sequences above is achieved when $r_{12} = r$. In this case each unit allocated across arc $\{12\}$ is at the expense of allocating a unit across each of arcs $\{11, 22\}$ (providing $w_1 = d_1$ and $w_2 = d_2$) for total reward $2r$. Therefore the maximum relative difference is

$$\frac{Z^* - Z}{Z^*} \leq \frac{r}{2r} = \frac{1}{2}.$$

Therefore the relative error due to applying the greedy algorithm to the recourse problem with a non-Monge sequence is potentially 50%. This underscores the potential need for exact solutions to the second-stage recourse problem.

An important assumption upon which the IDP was formulated is that inventory is allocated continuously. Although, in reality units of inventory are allocated in integer quantities. Since the recourse problem has network flow structure, if the right hand sides of the recourse problem constraints are integer valued then optimal second stage decision variables will be integer valued without the need for imposing explicit integrality restrictions (see proposition 2.3, Nemhauser and Wolsey, 1999, Chapter III.1, p. 541). Since d_i are integer by assumption, a sufficient condition is that that supplies, $U_j w_j$, are integer valued, which is true for the following two special cases:

i. (Deterministic Yield Loss): $U_j = u_j$, $\forall j$ w.p. 1, $\mathbf{w} \in Z_+^n$ and $\mathbf{d} \in Z_+^m$, or,

ii. (All or Nothing Yield Loss): U_j, $\forall j$ have distribution

$$U_j = \begin{cases} 1 & \text{w.p. } p_0 \\ 0 & \text{w.p. } 1 - p_0. \end{cases}$$

and $\mathbf{w} \in Z_+^n$ and $\mathbf{d} \in Z_+^m$.

For problems in which continuous allocation of inventory is an unacceptable approximation this permits a $O(K(nm + n + m))$ reduction in the number of integer decision variables.

However, in general the presence of stochastic yield loss ruins the unimodular structure of the constraint matrix by imposing non-integer right hand sides. Rather than concentrate on these special cases, we show instead that the assumption of continuous allocation of inventory is a reasonable approximation for the application considered in the Chapter.

Proposition 4 *At most c orders will receive partial (noninteger) allocation of supply.*

Proof: Every basis matrix of a transportation problem is triangular (see Corollary 13.2, pp. 382, Murty, 1983) and thus there exists an ordering of arcs such that greedy allocation (equivalently backward substitution) is optimal. Thus if it is optimal to allocate any supply from a supply node to a demand node then it is optimal to allocate the maximum possible supply. Since there are at most c supply sources it follows that at most c orders can receive only partial allocation of supply. □

Since d_i are typically large, and c is typically much smaller than m the relative worst case error from continuous allocation of inventory is expected to be small.

5.4 Heuristics

Solving the IDP is computationally difficult for two reasons: first because of its large scale, due to the stochastic nature of the problem, and second due to the additional combinatorial nature of the first-stage decision. As a result, exact solution, by branch-and-bound, for example, is not realistic. As discussed above there are similarities between the IDP and the well known p-median problem. The latter problem has been studied extensively and many different types of heuristics have been proposed for it (see Mirchandani and Francis, 1990, Chapter 2 for a review). Some of these heuristics can be extended in a straightforward manner to the IDP, however, the stochastic nature of the problem greatly increases their computational burden. We restrict the study to heuristics which are applicable to very large-scale inventory deployment problems such as those encountered in the steel industry.

We propose two straightforward heuristics. Each heuristic is based on iteratively determining design-choices, $\mathbf{x}$, such that at each iterations $\mathbf{x} \in B^n$. From proposition 2 of the previous section integrality restrictions for $\mathbf{q}$ can be dropped. Thus determination of $\mathbf{q}$, $\mathbf{w}$, $\mathbf{y}$, $\mathbf{e}$, $\mathbf{s}$, subject to the restriction on $\mathbf{x}$ reduces to solution of a two-stage stochastic linear program with continuous first and second stage decision variables. The first heuristic is a greedy type heuristic analogous to the one applied to the deterministic p-median problem in Chapter 4. We let J denote the chosen set of designs, i.e., $J = \{j \mid x_j = 1\}$, where $|J| \leq c$, and $Z^{LS}(J)$ is the greedy solution to the LSP.

Greedy Heuristic (GH):

Step 1: *Let* $\nu = 0$, $J = \emptyset$.

Step 2: *If* $\nu < c$, $v = v + 1$, *and let* $j_\nu = \arg\max_{j \notin J}\{Z^{LS}(J \cup j)\}$.

Step 3: *If* $Z^{LS}(J \cup j_\nu) \leq Z^{LS}(J)$ *then stop with* J *as greedy solution.*

Step 4: *If* $Z^{LS}(J \cup j_\nu) > Z^{LS}(J)$ *then* $J = J \cup j_\nu$. *Return to step 1.*

The critical difference between this heuristic and the dual-ascent heuristic used in Chapter 4 is that determination of j_ν at step 2 in iteration ν requires solution of $n - |J|$ instances of the LSP. The running time per iteration $O(n)$ in the number of potential choices of designs, however, computation times can be prohibitive depending on the size/computation time for the LSP.

The next heuristic is more suitable for very large size problems. It is based on initially solving a relaxation of the problem, followed by the solution of a restriction to the problem.

Decomposition Heuristic (DH):

Step 1 (Relaxation): *Relax the IDP to the corresponding deterministic p-median problem. Apply the greedy dual-ascent heuristic (Cornuejols, Fisher, Nemhauser, 1977) to determine the greedy solution,* $\mathbf{x}^d$.

Step 2 (Restriction): *Restrict the IDP to* $\mathbf{x} = \mathbf{x}^d$. *Solve the restricted IDP using the L-shaped method (Van Slyke and Wets, 1969) for* $\mathbf{q}^d$ *and* $\mathbf{w}^d$.

This heuristic is based on initially approximating the first-stage design-choice decision using the p-median relaxation of the problem. This results in a significant reduction in complexity by reducing the first-stage to an approximation (again using a greedy type heuristic) of a deterministic problem. The LSP is only solved once, after $\mathbf{x}$ has been specified. However, the choice of decision, $\mathbf{x}$, is made using only first moment information for $\boldsymbol{\xi}$.

5.5 Numerical Experiments and Empirical Observations

In this section we provide results from two computational studies. The first study consisted of numerical experiments to test the accuracy of the two heuristics proposed in the previous section by comparing the solutions obtained with the optimal solutions to small-size, randomly generated test problems. The second study involved a series of numerical examples based on empirical data from a particular ISM to demonstrate the economic importance of the model. The calculations in both studies were performed on a Sun Ultra 10 workstation with 128 MG Ram; the programming was done in C/C++. The commercial solver CPLEX was used both for solving mixed integer programs and linear programs.

5.5.1 Numerical Experiments

There is a wide range of possible problem structures that could arise in practice; the results presented here are for randomly generated test problems. A set of test problem sizes, specified by the choice of c, n, and m are examined. The problem instances for each of these

were generated as follows. From each of $i = 1, ..., m$ supply nodes, arcs were generated to each of $j = 1, ..., n$ demand nodes by sampling with probability p_i^a. Thus higher probabilities tend to correspond to greater substitutability of demand. The numerical experiments in Tables 5.1 and 5.2 both assume arc probabilities drawn from some distribution $F(p^a)$ where $p^a \subseteq [0, 1]$. Yield and demand were assumed to be i.i.d. in all cases and sampled according to $U_j \sim U(a, b), \forall j$ and $D_i \sim N(\mu, \sigma^2), \forall i$, to generate a set of K scenarios. Procurement costs $\mathbf{c}^p$ are assumed the same for all supply nodes and are fixed at 1. Coefficients for application rewards, shortage costs, and excess costs are all distributed as $U(1, 4)$.

$F(p^a)$	(c, n, m)	MV	ΔMV	LP	ΔLP	DH	ΔDH	GH	ΔGH
	(5,10,20)	9.43	15.65	0.26	1.21	0.31	2.09	4.73	8.57
	(5,10,30)	8.55	11.42	0.16	0.61	0.37	2.84	3.08	9.99
	(5,20,30)	2.77	4.67	1.01	2.89	1.88	7.53	6.34	7.53
$U(0.1, 0.3)$	(5,20,40)	7.05	11.03	1.69	4.43	0.61	2.96	4.70	11.01
	(10,20,30)	5.50	6.35	0.09	0.20	0.76	1.65	0.40	0.86
	(10,20,40)	5.29	6.77	0.12	0.35	0.49	1.02	0.31	0.77
	(10,30,50	3.59	4.79	0.12	0.31	0.18	0.47	1.80	2.94
	(5,10,20)	10.18	14.79	0.09	0.72	1.14	4.92	4.68	9.42
	(5,10,30)	9.40	15.4	0.04	0.35	0.11	0.66	3.30	6.30
	(5,20,30)	7.06	9.03	0.95	3.24	0.80	2.60	3.44	7.07
$U(0, 0.4)$	(5,20,40)	6.62	8.21	1.31	3.01	0.95	3.64	4.25	7.66
	(10,20,30)	6.06	7.19	0.03	0.09	0.47	1.21	0.16	0.51
	(10,20,40)	5.13	6.25	0.02	0.11	0.84	2.22	0.18	0.56
	(10,30,50	3.84	4.71	0.15	0.47	0.26	1.23	1.91	3.40

Table 5.1: Relative errors with respect to the optimum for randomly generated problem instances with $K = 25$, $U_j \sim U(0.8, 1), \forall j$, and $d_i \sim N(10, 2)$, $\forall i$, and $c_j^p = 0.5, \forall j$, $c_j^e = 2, \forall j$, $c_i^s = 2, \forall i$.

Within Tables 5.1 and 5.2 results are presented for different choices of arc probability distribution. The two tables differ with respect to the yield and demand uncertainty distributions; demand and yield variances are both higher in Table 5.1. The results illustrate the accuracy of DH and GH for small randomly generated test problems with respect to exact solutions obtained using the CPLEX mixed-integer-program solver. Relative errors

$F(p^n)$	(c,n,m)	MV	ΔMV	LP	ΔLP	DH	ΔDH	GH	ΔGH
	(5, 10, 20)	4.54	7.47	0.02	0.14	0.54	4.90	3.10	10.76
	(5, 10, 30)	3.11	4.92	0.49	3.05	0.80	3.75	2.72	7.50
$U(0.1, 0.3)$	(5, 20, 30)	2.64	4.51	0.89	2.12	0.79	4.01	4.80	7.97
	(5, 20, 40)	2.63	4.83	1.89	3.65	0.82	3.41	4.62	8.84
	(10, 20, 30)	2.84	3.72	0.03	0.12	0.36	0.64	0.24	0.56
	(10, 20, 40)	2.28	3.03	0.03	0.07	0.35	0.49	0.21	0.48
	(5, 10, 20)	4.92	6.11	0.03	0.12	0.29	2.15	2.95	6.11
	(5, 10, 30)	3.74	5.29	0.11	0.79	0.17	0.68	4.13	10.94
$U(0, 0.4)$	(5, 20, 30)	3.07	4.27	0.78	1.61	0.50	3.25	3.36	7.76
	(5, 20, 40)	2.60	3.57	0.88	3.27	1.08	5.59	4.78	8.40
	(10, 20, 30)	2.72	3.87	0.02	0.05	0.25	0.46	0.093	0.21
	(10, 20, 40)	2.48	3.10	0.03	0.15	0.31	0.57	0.17	0.61

Table 5.2: Relative errors with respect to the optimum for randomly generated problem instances equivalent to Table 5.1 except $U_j \sim U(0.85, 0.95), \forall j$, and $d_i \sim N(10, 1)$, $\forall i$

are reported as $100 \times (Z^* - Z)/Z$. In addition to the optimal solution, and heuristic solutions using GH and DH (lower bounds), the upper bound achieved by the linear program relaxation is reported in each case. Column headings in the tables are as follows:

MV: Average relative error for solution to mean-value p-median problem.

ΔMV: Maximum relative error for solution to mean-value p-median problem.

LP: Average relative error for the continuous linear programming relaxation.

ΔLP: Maximum relative error for the continuous linear programming relaxation.

DH: Average relative error for DH.

ΔDH: Maximum relative error for DH.

GH: Average relative error for GH.

ΔGH: Maximum relative error for GH.

Average and maximum errors were determined from solutions to 20 randomly generated problem instances in each case. Since DH and GH are heuristics they are lower bounds on

the optimal solution.

Results are very favorable for both heuristics DH and GH. The overall average error across all problem instances in Tables 5.1 and 5.2 (480 in total) was 0.65% for DH and 2.80% for GH. For DH the average error in Table 5.1 was 0.72% compared to 0.52% in Table 5.2. A similar reduction in average error for lower yield and demand variance of 2.96% to 2.6% was observed for GH. The worst case errors across all 480 test problems were 7.53% and 11.01% for DH and GH respectively. Also, in nearly 50% of the test problems DH found the optimal solution whereas GH succeeded in finding the optimum in about 12% of the cases. It is particularly interesting that DH in the majority of cases outperforms GH, even though the latter explicitly considers the impact of uncertainty through the solution of the LSP for each iteration of the design-choice decision. The degree to which DH is an improvement over GH appears to depend on the number of designs choices. When $c = 10$ the difference between the two heuristics is small compared to when $c = 5$. Intuitively this appears to imply that the performance of the GH heuristic is tied to the extent to which design substitution is possible.

Results in Tables 5.1 and 5.2 show that in general the continuous LP relaxation provides a tight upper bound on the optimum. The average gap is 0.48% in Table 5.1 and 0.45% in Table 5.2. The high quality of the bound from the LP relaxation has been observed for the deterministic p-median problem as well (Cornuejols et al., 1977). The solution of the p-median relaxation has an average relative error of 6.92% in Table 5.1 and 3.13% in Table 5.2. The p-median relaxation is more sensitive to changes in the variance of yield and demand than the LP relaxation.

Results in Table 5.3 are the same as those in Tables 5.1 and 5.2 except for differences in demand uncertainty and arc probabilities. In this case it is assumed that arc probabilities have a discrete distribution corresponding to a *truncated Pareto* distribution such that $P(10, 0.25) = C/k^{1.25}$, $k = 1, ..., 10$ and $C = [\sum_{k=1}^{10}(1/k)^{1.25}]^{-1}$. (Note that truncated

$F(p^a)$	(c,n,m)	MV	ΔMV	LP	ΔLP	DH	ΔDH	GH	ΔGH
	(5, 10, 20)	59.74	78.62	0.21	0.89	0.28	1.93	1.78	7.11
	(5, 10, 30)	41.36	58.55	0.04	0.44	0.00	0.00	1.00	3.54
	(5, 20, 30)	34.84	47.56	0.23	1.55	1.89	7.69	3.83	9.01
$P(10, 0.25)$	(5, 20, 40)	23.51	31.53	0.17	0.67	0.98	4.62	3.58	7.06
	(10, 20, 30)	33.46	50.91	0.04	0.11	0.77	2.08	0.51	1.19
	(10, 20, 40)	24.94	29.00	0.04	0.21	0.68	2.38	0.65	1.31
	(10, 30, 50	20.58	27.88	0.10	0.39	0.49	1.14	0.70	2.55

Table 5.3: Relative errors with respect to the optimum for randomly generated problem instances equivalent to Table 5.1 except $U_j \sim U(0.8, 1), \forall j$, and $d_i \sim N(10, 2)$ w.p. 0.5 and $d_i = 0$ w.p. 0.5, $\forall i$

Pareto was chosen because the results generated using it were roughly similar to those observed for empirical data.) Furthermore, demand for each of the order types is now assumed to by $N(10, 2)$ w.p. 0.5 and 0 w.p. 0.5. This allows for the additional uncertain as to whether an order-type (demand node) has nonzero demand. (In the next section we discuss how this type of uncertainty arises in the steel industry.) Results are again favorable for both heuristics in these numerical experiments as well. In this case average errors for DH and GH are 0.72% and 1.72%, respectively. However, it is interesting to note that the gap between the optimal solution and the upper bound from solving the p-median relaxation is much larger than in the examples in Tables 5.1, 5.2, and 5.3. This can be attributed to much greater demand variation resulting from significant probabilities of realizing zero demand for each order-type.

Figure 5.3 illustrates differences in average computation times for the exact solution, DH, and GH. It was found that relatively large problem instances could be solved exactly using the CPLEX mixed-integer solver. In fact all of the randomly generated samples could be solved in less than 60 minutes. Figure 5.3 is a comparison of average computation times over all test problems. From Figure 5.3 it is clear that DH has significantly shorter average computation times. In fact due to the computational burden of solving the LSP repeatedly in GH it shows only marginal improvement over solution to exact problems for these small

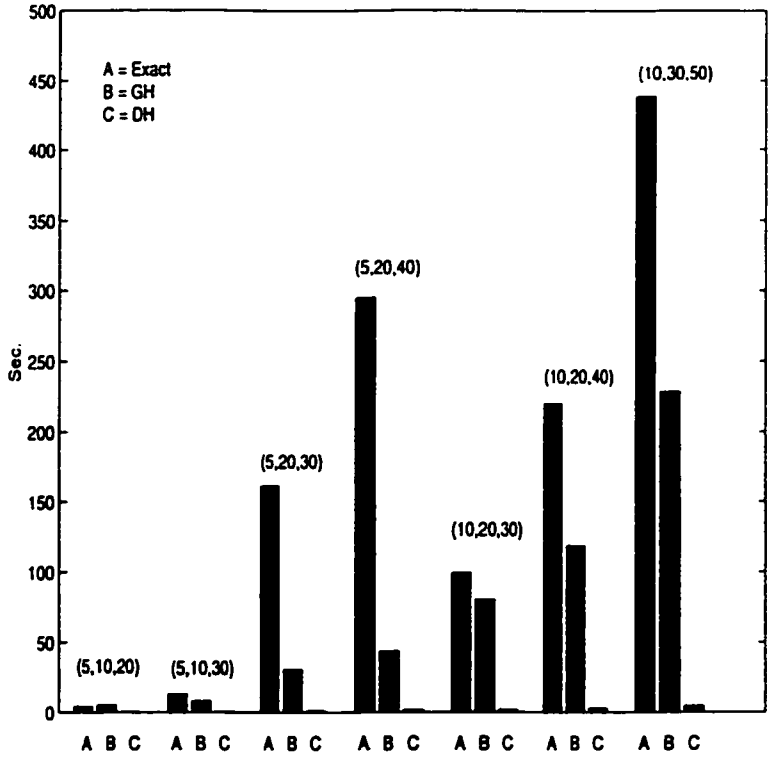

Figure 5.3: Average computation times in seconds net of problem setup time for exact solutions, and heuristics GH and DH.

test cases. In the next section we provide examples of its application to a real problem in the steel industry.

5.5.2 Empirical Example

The following is an application of the model to real-world problems faced by an ISM. The examples in this section illustrate the impact of uncertainty on lot-sizing decisions for an actual set of slab designs stocked at the ISM which were chosen based on solution to the p-median relaxation problem. **No specific information about the policies or design choices of the ISM involved in the study can be derived from the results below. However, the examples demonstrate the types of problems that can be**

studied using our model, as well as general trends and insights into the effects of uncertainty on inventory deployment decisions. We begin by explaining the method for generating problems using empirical order book data. The approach discussed below is based on discussions with senior managers at one ISM.

Design-choices, order-choices, and lot-size decisions, must be made in advance of knowing with certainty which orders will arise. These decisions are based on existing orders, which are typically placed sufficiently well in advance for planning purposes, however, there is uncertainty about possible changes to the size and/or timing of the release of orders between the time the planning decisions are made and the beginning of the shutdown. To simulate scenarios that are consistent with these two types of uncertainty two sets of orders are defined, a set of planned orders, I_p, and a set of possible *replacement* orders I_s. For instance, I_p might correspond to existing orders in the order book with processing time adjusted due dates during the planning period, and I_p would correspond to a set of likely replacements in the event of customer changes to due dates or cancellations. We define the set of order-types that arise in scenario k as I_p^k. Consistent with previous notation, the actual order size is d_i, and the order size in scenario k of order-type i is d_i^k. The demand scenarios are generated by the following 3-step process. (Note that the $+/-$ operations below correspond to addition and removal of elements from a set.)

Demand Scenario Generation:

For $k = 1, ..., K$ *do:*

Step 0: *Initialize* I_s *and* I_p *and set* $d_i^k = d_i, \forall i \in I_p$, *and* $I_p^k = I_p$.

Step 1: *For all* $i \in I_p^k$ *w.p.* p^s *perform the following interchange operations with a randomly drawn order,* i_c*:*

$$I_p = I_p - i + i_c, \qquad I_s = I_s + i - i_c.$$

Step 2: *For all* $i \in I_p$ *set* $d_i^k = (1 + \Delta)d_i$ *where* Δ *is a sampled random deviate satisfying* $\Delta > -1$.

Thus there are two parameters determining uncertainty in demand, p^s and Δ. To complete generation of a vector, ξ^k, random yields are sampled according to some appropriately defined yield distribution.

In practice the calibration of the various cost coefficients is a difficult problem. For the purpose of numerical examples considered here it is assumed that shortage costs are identical for all order-types, and excess and procurement costs are each the same for all slab designs. The application rewards, r_{ij}, depend on the relative differences between slab and order width and weight

$$r_{ij} = \begin{cases} r - c^w(w_j^s - w_i^d) + c^m(m_i^s - m_j^d) & \text{if order i is applicable to slab j} \\ 0 & \text{otherwise,} \end{cases}$$

where w_i^d, w_j^s are the widths for order i and slab j, respectively, and m_i^d, m_j^s, are the weights for order i and slab j, respectively. Coefficients c^w and c^m are calibrated with respect to the cold-application rules, such that the minimum reward for a feasible slab-to-order application satisfies $r_{ij} \geq 0$. For the purpose of generating scenarios, the parameters p_i^s and Δ_i were assumed the same for all order-types. Unless specified otherwise, it is assumed that yields are i.i.d. uniformly distributed, and that the demand scenario generation parameter Δ is distributed as $\sim N(\mu, \sigma^2)$. The planning period is such that the number of orders $\approx 10^3$.

c/K	25	100	300	500
5	3	18	67	112
10	23	92	281	480
15	33	190	635	1251

Table 5.4: Computation times net of problem setup time for solution of LSP.

Some limited examples of solution times are illustrated in Table 5.4. Experiments were performed with implementations of the algorithm with and without the use of the special

CPLEX network solver option. The sample computation times illustrate the fact that when using the network solver option, dependence on K for fixed n is roughly linear for decomposition algorithms. Dependence on n for fixed K, on the other hand, is approximately quadratic. It was found that solution times were roughly independent of changes to cost coefficients. Taking advantage of the network structure made solution times between 4 and 8 times faster for the problems considered in Table 5.4.

The sampling method described above is based on a crude Monte Carlo simulation of orders and order sizes. Statistical error depends on the number of scenarios as $1/\sqrt{K}$. Numerical experiments indicate that the 95% confidence intervals around the mean for $k = 50$ to $k = 500$ correspond to relative errors of roughly 0.8% to 8%. Thus, small statistical errors can be achieved for reasonable problem sizes without resorting to variance reduction methods like *importance sampling* (e.g. Infanger, 1994). In general it was found that the sample variance is relatively insensitive to changes in the yield variance, and more sensitive to the choice of scenario generation parameters, p^s, Δ.

Figures 5.4 and 5.5 illustrate the effects of yield uncertainty on the optimal solution. Figure 5.4 is a plot of the optimal solution, Z^*, as a function of yield variance, with fixed mean. Yields are assumed uniformly distributed as $U_j \sim U(0.8 - \delta/2, 0.8 + \delta/2), \forall j$, and Z^* is plotted against δ. When yields are deterministic, yield rate affects optimal lot-sizes as $w_j^* \rightarrow w_j^*/u_j, \forall j$. Thus, there is significant sensitivity to the first moment of the U_j when procurement costs, c_j^p, are high. Our numerical experimentation shows that Z^* can also be sensitive to changes in the second moments of the yield distribution. For example, in Figure 5.4, Z^* is decreasing at an increasing rate. For $\delta > 0.1$, Z^* is roughly linear in δ. This sensitivity to the second moment of the yield distribution is evidence that the mean-value solution would be a poor approximation when yield variance is high.

Figure 5.5 illustrates the difference between two different types of yield. The upper line

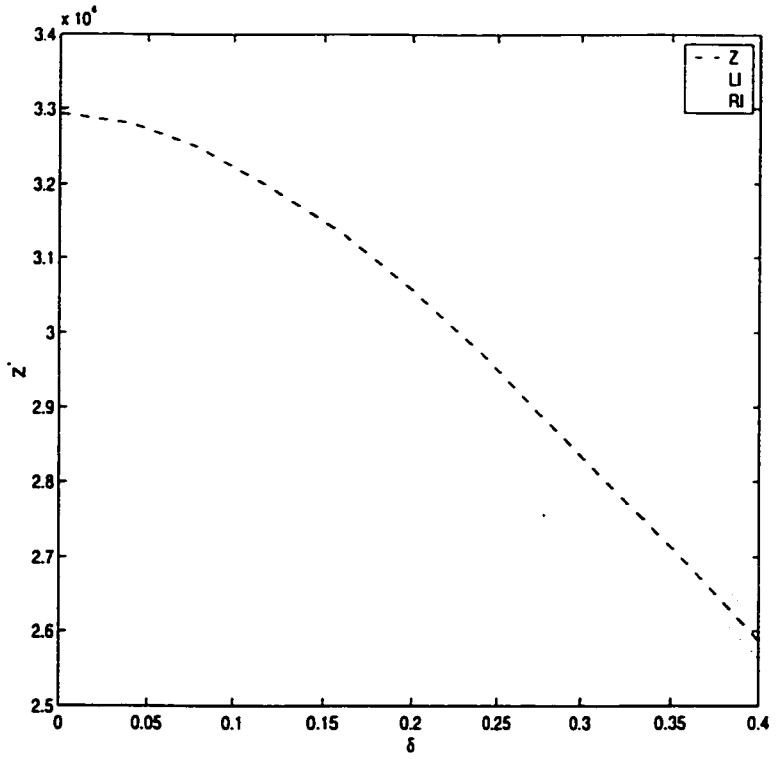

Figure 5.4: Z^* as a function of $\delta = (b-a)$ for $U_j \sim U(a,b)$ for $n = 15$, $K = 500$, $p^s = 0.1$, $\Delta \sim N(0, 0.1)$

corresponds to uniformly distributed yields, whereas the lower line corresponds to all-or-nothing type losses, i.e., $U_j = 1.0$ w.p. p_0 and $U_j = 0$ w.p. $1 - p_0$. The figure plots Z^* with respect to p_0. For the uniformly distributed case, $U_j \sim U(a,b), \forall j$, parameters are chosen such that the mean and variance are equivalent to the all-or-nothing case. (This is done to demonstrate that dependence on yield uncertainty goes beyond first and second moments of the distribution.) This is achieved by choosing parameters (a,b) for the uniformly distributed case such that $\frac{a+b}{2} = p_0$ and $\frac{(a-b)^2}{12} = p_0(1-p_0)$, corresponding to fixing means and variances respectively. The distinct differences in Z^* are therefore due to dependence on third and higher moments of the yield distribution.

Several numerical examples are presented in Tables 5.5 - 5.7 for different choices of cost

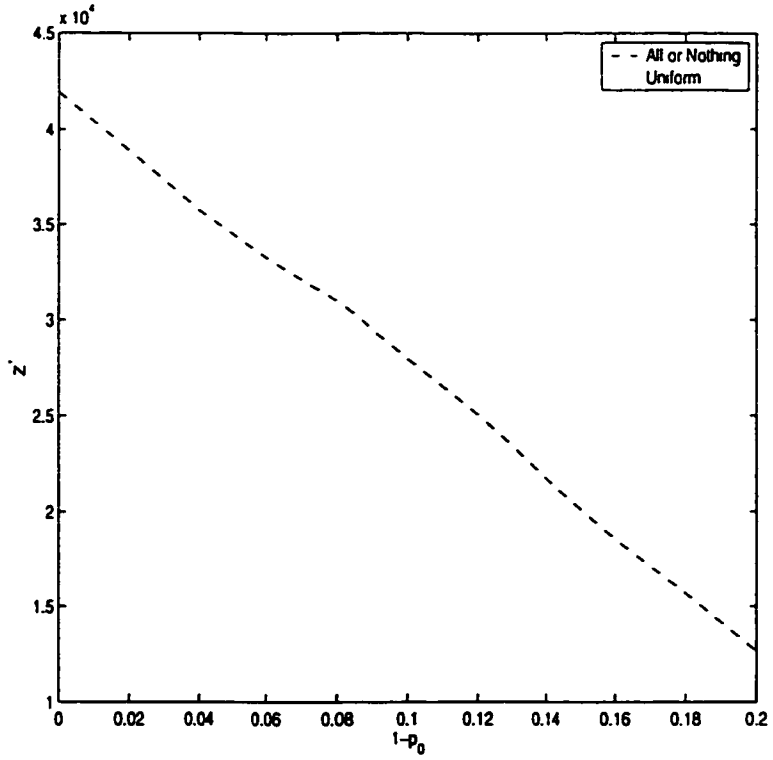

Figure 5.5: Z^* with respect to systematic changes in mean and variance for two types of yield distributions for $n = 15$, $K = 500$, $p^s = 0.1$, $\Delta \sim N(0, 0.1)$

coefficients and demand scenario parameters p^s and Δ (for simplicity yields are assumed perfect in these examples). In table 5.5 it is assumed that there is uncertainty in the types of orders ($p^s = 0.1$), but no uncertainty in the sizes of orders ($\Delta = 0$). In table 5.6 the opposite is assumed, the order-types are known with certainty ($p^s = 0$) but order sizes are uncertain ($\Delta \sim N(0, 0.1)$). Finally, in table 5.7 the two types of uncertainty are combined. From inspection of Tables 5.5 - 5.7 it is clear that Z^* is more sensitive to uncertainty in order-types than order sizes. The expected value of perfect information (EVPI) is higher in Table 5.5 (10% - 20%), than in Table 5.6 (about 2% - 13%). In Table 5.7, for which the two types of uncertainty are combined, the EVPI is higher than in either Tables 5.5 or 5.6, but less than the sum of the EVPIs from those two tables. Although the EVPI is significant

our results also indicate that the mean-value solution is often near optimal (within 1% to 6%).

Tables 5.8 - 5.10 contain results for several examples that illustrate relative differences between the optimal and mean-value lot-sizes for varying yield uncertainty. The mean-value lot-sizes are the optimal lot-sizes, given that yields and demand for order-types are deterministic and equal to their means. It is useful to make this comparison since this is the approach typically used in practice for choosing lot-sizes at the ISM. The most interesting observation is perhaps that total inventory is relatively insensitive to significant changes in cost parameters and yield uncertainty. Although intuitively it is expected that lot-size would increase with increasing yield variance, the dependence is found to be very weak. However, it was found that there may be significant dependence of individual lot-sizes on yield uncertainty (see x_7/μ_7 in Tables 5.8 - 5.10, for example).

5.6 Summary and Conclusions

The model presented in this Chapter is a first step in capturing aspects of inventory deployment under uncertainty that are relevant to the steel industry and other process industries. The numerical examples illustrate several potential uses for the model as a planning tool. While specific results depend on the particular application, the following general strategic-level conclusions seem warranted:

i. Choosing designs based on mean demand and yield typically yields optimal or near optimal design choices.

ii. EVPI is, in general high, indicating significant advantages associated with improving demand information within the supply chain (e.g. vendor managed inventories).

iii. Uncertainty in order-types (customer classes) has a greater impact on profits than uncertainty in order sizes.

iv. Total inventory is insensitive to yield uncertainty; however, individual lot-sizes can be strongly effected.

The model studied in this Chapter is applicable to general problems involving the design/configuration of transportation networks given uncertainty in supply and demand. The model could be modified to account for various other factors such as capacity constraints on total production and/or external purchases, or randomness in second-stage cost coefficients. However, suitability of the proposed heuristics for such problems remains to be established. The scenario generation approach described has the benefit of being relatively simple to automate with respect to a firm's changing order book. However, a potentially valuable extension would be the testing of the sensitivity of inventory deployment decisions to alternative *stress scenarios*. An example would be the sensitivity of decisions to significant changes in the order book, such as the addition of new products and/or customers.

c^p, c^s, c^e)	Z	σ_Y	LI	RI	WS	MV	$EVPI$
(1, 4, 4)	59008.2	87.9	58835.8	59180.6	64708.4	58192.3	5700.1
(2, 4, 4)	41684.0	87.1	41513.2	41854.7	47114.2	40598.1	5430.2
(3, 4, 4)	24508.2	87.9	24335.8	24680.7	29520.0	23004.0	5011.7
(1, 4, 2)	59929.0	81.9	59768.3	60089.6	64708.4	59227.9	4779.3
(2, 4, 2)	42275.6	80.9	42116.9	42434.2	47114.2	41633.8	4838.6
(3, 4, 2)	24924.1	84.1	24759.2	25088.9	29520.0	24039.6	4595.9
(1, 4, −0.5)	62663.8	71.0	62524.5	62803.0	64708.4	60522.5	2044.5
(2, 4, −0.5)	43795.8	68.1	43662.3	43929.3	47114.2	42928.3	3318.3
(3, 4, −0.5)	25812.6	73.2	25669.1	25956.2	29520.0	25334.2	3707.4

Table 5.5: Optimal solutions, confidence intervals and other numerical results for 15 slab designs, no yield losses, $K = 500$, $p^s = 0.1$, and $\Delta = 0$

c^p, c^s, c^e)	Z	σ_Y	LI	RI	WS	MV	$EVPI$
(1, 4, 4)	57426.6	39.9	57348.3	57505.0	59272.4	57005.4	1845.7
(2, 4, 4)	41262.9	36.1	41192.1	41333.8	43165.4	40898.4	1902.4
(3, 4, 4)	29829.5	107.8	29618.0	30040.9	34401.4	29503.9	4571.9
(1, 4, 2)	58172.1	47.1	58079.6	58264.6	59708.1	57557.5	1536.0
(2, 4, 2)	41792.2	34.4	41724.6	41859.7	43518.3	41443.7	1726.0
(3, 4, 2)	27411.6	34.6	27343.7	27479.4	29507.0	27077.9	2095.4
(1, 4, −0.5)	64824.6	36.4	64753.0	64896.1	65300.6	63655.5	476.0
(2, 4, −0.5)	46857.9	28.5	46802.0	46913.7	47754.5	46140.9	896.6
(3, 4, −0.5)	21863.5	30.9	21802.8	21924.1	23422.2	21507.6	1558.6

Table 5.6: Optimal solutions, confidence intervals, and other numerical results for 15 slab designs, no yield losses, $K = 500$, $p^s = 0$, and $\Delta \sim N(0, 0.1)$.

c^p, c^s, c^e)	Z	σ_Y	LI	RI	WS	MV	$EVPI$
(1, 4, 4)	58437.2	96.5	58248.0	58626.4	64645.1	57677.1	6207.8
(2, 4, 4)	41012.3	92.4	40831.2	41193.5	47068.7	40100.7	6056.4
(3, 4, 4)	29204.8	141.8	28926.7	29482.9	36795.6	28238.8	7590.8
(1, 4, 2)	60375.8	97.4	60184.7	60566.8	65636.1	59474.7	5260.3
(2, 4, 2)	42181.1	82.0	42020.3	42342.0	47643.7	41520.8	5462.5
(3, 4, 2)	26590.3	92.8	26408.3	26772.3	32094.5	25773.4	5504.1
(1, 4, −0.5)	68584.7	82.0	68424.0	68745.4	70625.0	66205.2	2040.3
(2, 4, −0.5)	48495.0	75.0	48347.9	48642.2	51969.1	47368.2	3474.0
(3, 4, −0.5)	21706.1	76.2	21556.8	21855.5	25449.5	21278.0	3743.3

Table 5.7: Optimal solutions, confidence intervals, and other numerical results for 15 slab designs, no yield losses, $K = 500$, $p^s = 0.1$, and $\Delta \sim N(0, 0.1)$.

c^p, c^s, c^e)	$\frac{x_1^*}{\mu_2}$	$\frac{x_2^*}{\mu_2}$	$\frac{x_3^*}{\mu_3}$	$\frac{x_4^*}{\mu_4}$	$\frac{x_5^*}{\mu_5}$	$\frac{x_6^*}{\mu_6}$	$\frac{x_7^*}{\mu_7}$	$\frac{x_8^*}{\mu_8}$	$\frac{x_9^*}{\mu_9}$	$\frac{x_{10}^*}{\mu_{10}}$	$\frac{\sum x_j^*}{\sum \mu_j}$
(1, 4. 4)	0.99	0.99	0.95	0.90	1.03	0.98	1.86	0.93	0.97	0.89	1.00
(2. 4. 4)	0.96	0.98	0.96	0.91	1.23	0.94	1.11	0.94	1.48	0.87	0.99
(3, 4, 4)	0.95	0.96	0.95	0.89	1.21	0.92	1.08	0.92	1.46	0.85	0.97
(1, 4, 2)	1.02	1.02	0.98	0.96	1.07	1.12	1.32	0.93	1.04	1.02	1.02
(2, 4, 2)	1.00	1.00	0.93	0.92	1.04	1.01	1.24	0.95	0.88	0.86	0.99
(3, 4, 2)	0.97	0.97	0.93	0.95	1.03	1.08	1.64	0.90	0.99	0.74	0.97
(1, 4, −0.5)	1.07	1.13	1.08	1.09	1.45	1.67	1.28	1.18	1.19	1.48	1.12
(2, 4, −0.5)	1.01	1.08	0.96	1.06	1.18	1.04	1.14	1.09	1.17	1.06	1.05
(3, 4, −0.5)	0.98	1.02	0.96	0.95	1.19	1.02	1.12	1.01	1.27	0.73	1.01

Table 5.8: Lot-sizes with respect to mean-value lot-sizes for 10 slab designs, $U_j \sim U(0.9, 0.9), \forall j$, and $K = 500$, $p^s = 0.1$, and $\Delta \sim N(0, 0.1)$.

c^p, c^s, c^e)	$\frac{x_1^*}{\mu_2}$	$\frac{x_2^*}{\mu_2}$	$\frac{x_3^*}{\mu_3}$	$\frac{x_4^*}{\mu_4}$	$\frac{x_5^*}{\mu_5}$	$\frac{x_6^*}{\mu_6}$	$\frac{x_7^*}{\mu_7}$	$\frac{x_8^*}{\mu_8}$	$\frac{x_9^*}{\mu_9}$	$\frac{x_{10}^*}{\mu_{10}}$	$\frac{\sum x_j^*}{\sum \mu_j}$
(1, 4, 4)	1.00	0.99	0.94	0.91	1.03	0.99	1.89	0.92	0.97	0.89	1.01
(2, 4, 4)	0.95	0.98	0.94	0.90	1.25	0.94	1.23	0.94	1.46	0.87	0.99
(3, 4, 4)	0.94	0.96	0.94	0.88	1.23	0.92	1.14	0.92	1.44	0.84	0.98
(1, 4, 2)	1.02	1.02	0.97	0.96	1.07	1.10	1.32	0.93	1.04	1.03	1.03
(2, 4, 2)	1.00	1.00	0.91	0.91	1.04	1.00	1.28	0.95	0.87	0.87	1.00
(3, 4, 2)	0.97	0.97	0.92	0.95	1.02	1.08	1.73	0.90	0.98	0.75	0.98
(1, 4, −0.5)	1.06	1.13	1.06	1.08	1.43	1.58	1.27	1.17	1.18	1.40	1.14
(2, 4, −0.5)	1.01	1.07	0.95	1.05	1.17	1.04	1.16	1.07	1.17	1.05	1.05
(3, 4, −0.5)	0.97	1.01	0.94	0.94	1.23	1.01	1.15	0.99	1.25	0.74	1.02

Table 5.9: Lot-sizes with respect to mean-value lot-sizes for 10 slab designs, $U_j \sim U(0.85, 0.95), \forall j$, and $K = 500$, $p^s = 0.1$, and $\Delta \sim N(0, 0.1)$

c^p, c^s, c^e)	$\frac{x_1^*}{\mu_2}$	$\frac{x_2^*}{\mu_2}$	$\frac{x_3^*}{\mu_3}$	$\frac{x_4^*}{\mu_4}$	$\frac{x_5^*}{\mu_5}$	$\frac{x_6^*}{\mu_6}$	$\frac{x_7^*}{\mu_7}$	$\frac{x_8^*}{\mu_8}$	$\frac{x_9^*}{\mu_9}$	$\frac{x_{10}^*}{\mu_{10}}$	$\frac{\sum x_j^*}{\sum \mu_j}$
(1, 4, 4)	1.00	1.01	0.91	0.91	1.03	1.01	2.25	0.93	0.98	0.89	1.01
(2, 4, 4)	0.94	0.99	0.90	0.91	1.35	0.94	1.54	0.94	1.42	0.87	0.99
(3, 4, 4)	0.92	0.96	0.88	0.89	1.33	0.92	1.51	0.92	1.40	0.85	0.97
(1, 4, 2)	1.03	1.04	0.98	0.96	1.08	1.11	1.33	0.93	1.05	1.04	1.04
(2, 4, 2)	1.01	1.01	0.89	0.91	1.04	1.02	1.39	0.97	0.88	0.88	1.00
(3, 4, 2)	0.96	0.98	0.91	0.96	1.03	1.11	1.79	0.88	0.99	0.76	0.98
(1, 4, −0.5)	1.09	1.15	1.09	1.09	1.44	1.59	1.30	1.18	1.20	1.41	1.15
(2, 4, −0.5)	1.02	1.09	0.94	1.06	1.18	1.05	1.22	1.08	1.17	1.06	1.06
(3, 4, −0.5)	0.98	1.02	0.94	0.94	1.28	1.01	1.19	1.01	1.22	0.75	1.02

Table 5.10: Lot-sizes with respect to mean-value lot-sizes for 10 slab designs, $U_j \sim U(0.8, 1.0), \forall j$, and $K = 500$, $p^s = 0.1$, and $\Delta \sim N(0, 0.1)$.

Chapter 6

Summary and Extensions

This Chapter summarizes the work presented in this dissertation, underscores important insights arising out of this work, and suggests potential future extensions. The discussion is separated into two parts. The first considers the appointment scheduling problem from Chapter 3, the second the inventory deployment problems in Chapters 4 and 5.

6.1 Appointment Scheduling Systems

The work in Chapter 3 consists of a detailed study of a model applicable to the problem of optimizing appointment-based service systems. The model assumes a set of jobs scheduled on a stochastic server. The scheduling problem is a convex minimization problem, and solution of that problem yields job allowances (equivalently, appointment times) that minimize the combined costs of waiting time, idle time, and tardiness. A variation of the standard L-shaped algorithm is used to obtain approximate, but near-optimal, solutions via successively finer partitions of the support of job durations. Also, we derive upper and lower bounds on the accuracy of the approximation with respect to the optimal solution. For situations where jobs can be sequenced arbitrarily, we develop a bound on the magnitude of savings that can be realized from an optimal sequence of jobs. We show that if job-duration

variances increase, while keeping their means fixed, the expected total costs of running an appointments-based service system also increase concomitantly. We propose and test two easy-to-implement heuristics that are based on relaxations of the original problem. These heuristics are very fast and reasonably accurate for problems with a large number of jobs.

Several numerical experiments are presented which illustrate computation times, and dependence of solutions with respect to parametric variation of input parameters. Optimal appointment schedules are shown to be very sensitive to changes in the first and second moments of the job duration distributions. Several examples are given for different cost structures. Solutions are categorized according to the difference between waiting and idling cost coefficients. For instance, a high ratio of waiting to idling cost resulted in high sensitivity of the objective function to changes in the optimal schedule. However, when the converse is true the objective function is found to be relatively insensitive to changes. In fact, the *mean-value solution* (the solution obtained by setting job allowances equal to the first moments of job durations) is typically near optimal in this case. A bound on the benefits of changing the sequence of jobs was derived. The major insight provided by this bound is that the dependence of the objective function on job sequence is high when the ratio of idling to waiting costs is high. When the opposite is true, the dependence is low. Results of the numerical experiments and the bound on changes in the objective function with respect to job sequence have important implications for managers of appointments-based service systems. They serve to categorize the types of systems for which application of a stochastic optimization model could have significant impact on total costs.

There are several potential extensions to the appointment scheduling problem which to the authors knowledge have not yet been studied. These can be classified into three groups. The *first* is a more detailed analysis of the effects of job sequence. This problem is combinatorial in nature, and can be described as follows. Given that optimal job allowances are used for any particular sequence of jobs, the problem is to determine a sequence that minimizes

total costs. In the simple case of i.i.d. job durations, and identical waiting and idling cost coefficients for all jobs, the total cost of operating the system is obviously independent of job sequence. However, if either job durations or waiting/idling cost coefficients are job-dependent, there can be significant improvements from determining the optimal sequence. Furthermore, there is evidence that the optimal solution to the sequencing problem can be counter-intuitive. For example, Wang (2000) conjectures that the increasing-variance ordering of jobs is optimal. Ridder et al. (1998) give a two job example in the context of a news-vendor problem that proves the opposite (decreasing-variance) ordering can be optimal. In particular, they show that there exist probability distributions for which higher variance may be preferred to lower variance. That is consistent with a decreasing-variance ordering in the two job appointment scheduling examples, and inconsistent with Wang's conjecture.

The *second* potential extension deals with general project scheduling environments in which the predecessors for a given job may be more arbitrarily defined. In the appointment scheduling problem discussed in Chapter 3, the jobs were performed sequentially. However, in typical project scheduling applications (e.g. building construction, product design) start times of jobs may depend on completion of multiple predecessors, and some jobs may be performed in parallel. As in the appointment scheduling problem typical costs in project scheduling are due to idling of resources and total tardiness of the project with respect to some fixed completion time. The more general project scheduling problem can also be formulated as a two-stage stochastic linear program, however, the much larger size of problems encountered in project scheduling applications requires sampling based approaches.

The *third* extension would consider the case of more general queuing networks. Arriving customers may require visits to multiple types of servers. Furthermore, there may be multiple, but identical, servers providing each service. In such cases the optimal sequence of servers visited by a customer may depend on availability at the time of arrival and

during service. This results in the added complexity of optimizing the routing of customers through the system on a dynamic basis. Such models would allow insights into the effects of varying the configuration of servers in more complex appointments based service systems (e.g. triage systems at outpatient clinics, multiple operating room scheduling).

Together, the above extensions would improve understanding of how appointment scheduling policies, and the configuration of project networks and queuing networks, affect the waiting of customers, and the idling of resources within appointment based service systems. Such information would improve daily scheduling decisions and job/server sequencing and workload allocations, as well as long term strategic decisions about capacity investments and system design.

6.2 Inventory Deployment in the Steel Industry

The term inventory deployment is used to describe the coordinated decision of choosing the design, order-types, and lot-sizes for planning inventory as part of a make-to-stock production mode in the steel industry. The work in Chapters 4 and 5 comprises a detailed study of models for semi-processed inventory deployment with special applicability to the steel industry (but not restricted to that industry). Chapter 4 considers the problem of determining a finite set of slab designs from a continuous range. It is shown that the continuous set of potential designs can be reduced to a finite set. The resulting finite problem is shown to be equivalent to a well known problem that has been studied previously in the context of facility location decisions. It is also shown, by numerical analysis, that fast heuristics can be used to provide very accurate results for this problem. The model has been implemented at a particular ISM, and details of its implementation were provided. Numerical examples using problem parameters typical in steel manufacturing are provided and these show evidence of the Pareto rule for semi-processed items in that industry. Furthermore, subsequent

to the work reported here, the model has been applied to the optimization of another type of semi-finished inventory (flat rolled coils referred to as *band*), with only minor modifications of its original form.

Chapter 5 extends the design problem of Chapter 4 to include the added complexity of planning lot-sizes for the chosen designs in the presence of uncertain demand and yield. The model captures the effects of random order-types, order-sizes, and yield rates. The structure of the problem is analyzed and several properties are presented which lay the groundwork for two heuristics. Average and worst-case performances of the heuristics are estimated, and shown to be quite favorable, by solving a series of randomly generated test problems. Several numerical examples are provided that demonstrate the impact of demand and yield uncertainty in inventory planning and managerial insights are summarized. In particular, it is shown that the solution to the mean-value problem typically results in near optimal design choices, however, there may be significant advantages to solving the stochastic optimization problem for determining lot-sizes. The expected value of perfect information is typically large which is evidence of the importance to integrated steel manufacturers of engaging in projects that increase information sharing through the supply chain (e.g. vendor-managed inventory). Also, total inventory is found to be relatively insensitive to yield uncertainty, however, individual lot-sizes may be significantly affected.

The models studied in Chapters 4 and 5 represent a first step in the study of large scale models for the strategic deployment of semi-processes inventory in the steel industry. An important and potentially valuable extension to these models is to the case of multi-period planning. The models we consider are of the two-stage type, however, in reality this is a simplification. In practice planners have choices to make about dynamic changes to the portfolio of inventory carried including: decisions to add additional designs, remove designs, and/or switch from one design type to another. Two-stage models are a natural first step in the study of models for dealing with the much larger and complex multi-period problems.

In the multi-period setting there is the added complexity of coordinating decisions among multiple time periods. The extremely large size of the problems and the need for discrete decisions for the choice of whether to choose/switch an item of inventory make even the implementation of reasonable heuristics a challenging goal.

The inventory deployment problem was motivated in the context of a slab inventory system for an integrated steel manufacturer. However, the models that were developed are generalizable, not only to other stages of semi-finished inventory in the steel industry (e.g. band inventory), and other process industries (e.g. pulp and paper), but also to other problem contexts. For example, in the context of long-term strategic management decisions, the problem is analogous to the choice of facility locations (in fact, the typical context for discussing p-median models). Similarly, the configuration of facilities with regard to flexible manufacturing capability in a multi-product setting (e.g. Graves, 1994) is an important and similar problem to which models of the kind we have developed can be applied.

Appendix A

A.1 Upper Bounding Lagrangian Dual Formulation

We begin by relaxing constraints (4.4) from the original formulation in (4.3) - (4.7) of section 4.4. For a given set of Lagrange multipliers, λ, a dual of the relaxed problem can be written as follows:

$$Z_D = \min_{\lambda}\{Z_D(\lambda)\}, \tag{A.1}$$

where

$$Z_D(\lambda) = \max_{x,y}\{L(x,y,\lambda)\}, \tag{A.2}$$

and the Lagrange function L is written as

$$L(x,y,\lambda) = \sum_{i=1}^{c}\sum_{j=1}^{s} r_{ij}y_{ij} + \sum_{i=1}^{c}\lambda_i(1 - \sum_{j=1}^{s} y_{ij}). \tag{A.3}$$

The above problem is to be solved subject to constraints (4.5) through (4.7) of the original formulation presented in section 4.4. It is well known that $Z_D \geq Z$. Cornuejols, Fisher and Nemhauser (1977) present arguments that the coefficient matrix of constraints (4.5) to (4.7) for given feasible vector $\mathbf{x}$ is totally unimodular, i.e., a linear programming (LP) relaxation of the problem of determining $Z_D(\lambda)$ yields integer y. In fact, due to the simple structure of $Z_D(\lambda)$ the problem can be solved without having to solve an LP, as shown in Cornuejols, Fisher, and Nemhauser (1977).

It is well known that $Z_D(\lambda)$ is non-differentiable, but piecewise linear and convex in λ. Therefore, Z_D can be determined efficiently via the sub-gradient method (see, for example, Nemhauser and Wolsey, 1999, pages 41-49 for details). When using the sub-gradient method, the user needs to specify a stopping criterion. In our implementation, the algorithm was terminated when M successive iterations produced a net improvement in Z_D which was less than an arbitrary ϵ. Normalized sub-gradients were used as search directions, and step sizes were decreased geometrically at regular step intervals. The step size was reduced by a factor of 0.98 every 5 iterations, M was set to 100 iterations, and the algorithm was terminated when the net improvement was less than 0.001%.

Bibliography

[1] Bahn, O., du Merle, O., Goffin, J.L. and Vial, J.P. 1995. A Cutting Plane Method from Analytic Centers for Stochastic Programming. *Mathematical Programming.* **69** 45-73.

[2] Bailey, N. 1952. A Study of Queues and Appointment Systems in Hospital Outpatient Departments, with Special Reference to Waiting-Times. *Journal of the Royal Statistical Society.* **A14** 185-189.

[3] Bailey, N. 1954. Queuing for Medical Care. *Applied Statistics.* **3** 137-145.

[4] Bassok, Y., Anupindi, R. and Akella, R. 2000. Single Period Multiproduct Inventory Models with Substitution. *Operations Research.* **47** 632-642.

[5] Beale, E.M.L. 1955. On Minimizing a Convex Function Subject to Linear Inequalities, *Royal Statistical Society, Series B.* **17** 173-184.

[6] Benders, J.F. 1962. Partitioning Procedures for Solving Mixed-Variable Programming Problems, *Numerische Mathematik.* **4** 238-252.

[7] Birge, J.R. 1980. Solution Methods for Stochastic Dynamic Linear Programs, Ph.D. Dissertation, Stanford University.

[8] Birge, J.R. 1980. Technical Report SOL80-29, Systems Optimization Laboratory, Stanford University.

[9] Birge, J.R. 1982. The Value of the Stochastic Solution in Stochastic Linear Programs with Fixed Recourse, *Mathematical Programming.* **24** 314-325.

[10] Birge, J.R. 1985. Decomposition and Partitioning Methods for Multistage Stochastic Linear Programs, *Operations Research.* **33** 989-1007.

[11] Birge, J.R. 1985. Aggregation Bounds in Stochastic Linear Programming. *Mathematical Programming.* **31** 25-41.

[12] Birge, J.R. and Wallace, S.W. 1986. Refining Bounds for Stochastic Linear Programs With Linearly Transformed Independent Random Variables, *Operations Research Letters.* **5** 73-77.

[13] Birge, J.R., Wets, R.J-B. 1986. Designing Approximation Schemes for Stochastic Optimization Problems, in Particular, for Stochastic Programs with Recourse, *Mathematical Programming Study.* **27** 54-102.

[14] Birge, J.R., Qi, L. 1988. Computing Block Angular Karmarkar Projections with Applications to Stochastic Programming, *Management Science.* **34** 1472-1479.

[15] Birge, J.R. and Louveaux, F.V. 1988. A Multicut Algorithm for Two-Stage Stochastic Linear Programs, *European Journal of Operational Research.* **34** 384-392.

[16] Birge, J.R and Maddox, M.J. 1995. Bounds on Expected Project Tardiness, *Operations Research.* **5** 838-850.

[17] Birge, J.R. 1995. Current Trends in Stochastic Programming Computation and Applications, *University of Michigan Working Paper.* 48109-2117.

[18] Birge, J.R. and Louveaux, F. 1997. *Introduction to Stochastic Programming*, Springer-Verlag, New York.

[19] Bitran, G.R. and Dasu, S. 1992. Ordering Policies in an Environment of Stochastic Yields and Substitutable Demands. *Operations Research.* **40** 999-1017.

[20] Brahimi, M. and Worthington, D.J. 1991. Queuing Models for Out-patient Appointment Syste ms-a Case Study. *Journal of the Operational Research Society.* **42** 733-746.

[21] Carino, D.R. and Ziemba, W.T. 1998. Formulation of the Russell-Yasuda Kasai Financial Planning Model. *Operations Research.* **46** 433-449.

[22] Carino, D.R., Myers D.H. and Ziemba W.T. 1998. Concepts, Technical Issues and Uses of the Russell-Yasuda Kasai Financial Planning Model. *Operations Research.* **46** 450-462.

[23] Caroe, C., Ruszczynski, A. and Schultz, R. 1997. Unit Committment Under Uncertainty via Two-Stage Stochastic Programming. Technical Report, Department of Computer Science, University of Copenhagen, Number DIKU-TR-97/23.

[24] Carpenter, T., Lustig, I. and Mulvey, J. 1991. Formulating Stochastic Programs for Interior Point Methods, *Operations Research.* **39** 757-770.

[25] Charnes, A. and Cooper, W.W. 1959. Chance Constrained Programming, *Management Science.* **5** 73-79.

[26] Charnetski, J. 1984. Scheduling Operating Room Surgical Procedure with Early and Late Completion Penalty Costs. *Journal of Operations Management.* **5** 91-102.

[27] Choi, I.C. and Goldfarb, D. 1993. Exploiting Special Structure in a Primal-Dual Path-Following Algorithm. *Mathematical Programming.* **58** 33-52.

[28] Chen, F. 2000. Quantifying the Bullwhip Effect in a Simple Supply Chain: The Impact of Forcasting, Lead Times, and Information. *Management Science.* **46** 436-444.

[29] Cornuèjols, G., Fisher, M.L. and Nemhauser, G.L., 1977. Location of Bank Accounts to Optimize Float. *Management Science.* **23** 789-810.

[30] Dantzig, G.B. 1955. Linear Programming Under Uncertainty, *Management Science.* **1** 197-206.

[31] Dantzig, G.B. and Wolfe, P. 1960. The Decomposition Principle for Linear Programs, *Operations Research.* **8** 110-111.

[32] Dantzig, G.B. and Glynn, P. 1990. Parallel Processors for Planning Under Uncertainty, *Annals of Operations Research.* **22** 1-21.

[33] Dempster, M.A.H. 1986. *Stochastic Programming*, Academic Press, New York.

[34] Dexter, F. 1999. Design of Appointment Systems to Minimize Patient Waiting Times: A Review of Computer Simulation and Patient Survey Studies. *Anesthesia and Analgesia.* **89** 925-931.

[35] Dietrich, B.L. Monge Sequences, Antimatroids, and the Transportation Problem with Forbidden Arcs. *Linear Algebra and its Applications.* **139** 133-145.

[36] Eeckhoudt, L.G.C. and Schlesinger, H. 1995. The Risk Averse (and Prudent) Newsboy. *Management Science.* **41** 786-794.

[37] Eppen G.G., Martin, R.K. and Schrage, L. 1989. A Scenario Approach to Capacity Planning, *Operations Research.* **37** 517-527.

[38] Ermoliev, Y and Wets, R.J-B. 1988. Stochastic Programming, an Introduction, in: Ermoliev, Y. and Wets, R.J.B. (eds.): *Numerical Techniques for Stochastic Optimization*, Springer-Verlag, Berlin.

[39] Ermoliev, Y, 1988. Stochastic Quasigradient Methods, in: Ermoliev, Y. and Wets, R.J-B. (eds.): *Numerical Techniques for Stochastic Optimization*, Springer-Verlag, Berlin.

[40] Ferguson, A. and Dantzig, G.B. 1956. The Allocation of Aircraft to Routes: An Example of Linear Pgramming Under Uncertain Demands, *Management Science.* **3** 45-73.

[41] Forst, F.G. 1993. Stochastic Scheduling on One Machine with Earliness and Tardiness Penalties. *Probability in Engineering and Informational Sciences.* **7** 291-300.

[42] Frauendorfer, K. and Kall, P. 1988a. A Solution Method for SLP Recourse Problems with Arbitrary Multivariate Distributions-The Independent Case, *Problems in Control and Information Theory.* **17** 177-205.

[43] Frauendorfer, K. 1988b. Solving S.L.P. Recourse Problems with Arbitrary Multivariate Distributions - The Dependent Case, *Mathematics of Operations Research.* **13** 377-394.

[44] Frauendorfer, K. 1992. Stochastic Two-Stage Programming, Lecture Notes in Economics and Mathematical Systems 392, Springer-Verlag, Berlin.

[45] Gartska, S.J. 1980. An Economic Interpretation of Stochastic Programs, *Mathematical Programming.* **18** 62-67.

[46] Gassman, H.I. 1990. MSLiP: A Computer Code for the Multistage Stochastic Lienar Programming Problem, *Mathematical Programming.* **47** 407-423.

[47] Geoffrion, A.M. 1970. Elements of Large Scale Mathematical Programming, *Management Science.* **11** 652-691.

[48] Gerchak, Y., Tripathy, A. and Wang, K. 1996. Coproduction Models with Random Functionality Yields. *IIE Transactions.* **28** 391-403.

[49] Goldman, J., H.A. Knappenberger and W.J. Shearson 1970. A Study of the Variability of Surgical Estimates. *Hospital Management.* **110** 46-46D.

[50] Hakimi, S.L. 1964. Optimum Location of Switching Centers and the Absolute Centers and Medians of a Graph. *Operations Research.* **12**, 450-459.

[51] Henig, M. and Gerchak, Y. 1990. The Structure of Periodic Review Policies in the Presence of Random Yield. *Operations Research.* **38** 634-643.

[52] Higle, J. and Sen, S. 1991. Stochastic Decomposition: An Algorithm for Two-Stage Linear Programs with Recourse, *Mathematics of Operations Research.* **16** 650-669.

[53] Ho, J.K. and Manne, A.S. 1974. Nested Decomposition for Dynamic Models, *Mathematical Programming.* **6** 121-140.

[54] Ho, C.-J. and Lau, H.-S., 1992. Minimizing Total Cost in Scheduling Outpatient Appointments. *Management Science.* **38** 1750 - 764.

[55] Hsu, A., and Bassok, Y. 1999. Random Yield and Random Demand in a Production System with Downward Substitution. *Operations Research.* **47** 277-290.

[56] Huang, C.C., Ziemba, W.T. and Ben-Tal, A. 1977. Bounds on the Expectation of a Convex Function o f a Random Variable: With Applications to Stochastic Programming. *Operations Research.* **25** 315-325.

[57] Ignall, E. and Veinott, A. 1969. Optimality of Myopic Inventory Policies for Several Substitute Products, *Management Science.* **15** 284-304.

[58] ILOG Inc., CPLEX Division, 1998. CPLEX Installation and Use Notes, Incline Village, NV.

[59] Infanger, G. 1992. Monte Carlo (Importance) Sampling Within A Benders Decomposition Algorithm for Stochastic Linear Programs, *Annals of Operations Research.* **39** 69-95.

[60] Infanger, G. 1994. *Planning Under Uncertainty: Solving Large-Scale Stochastic Linear Programs*, Boyd and Fraser, Danvers, Massachusetts.

[61] Jansson, B. 1966. Choosing a Good Appointment System - A Study of Queues of the Type (D,M,1). *Operations Research.* **14** 292-312.

[62] Jucker, J.V. and Rosenblatt, M.J. 1985. Single Period Inventory Models with Demand Uncertainty and Quantity Discounts: Behavioral Implications and a New Solution Procedure, *Naval Research Logisitics Quarterly.* **32** 537-550.

[63] Kall, P. 1979. Computatonal Methods for Solving Two-Stage Stochastic Linear Programming Problems, *Journal of Applied Mathematics and Physics.* **30** 261-271.

[64] Kall, P., Ruszczyński, A. and Frauendorfer, K. 1988. Approximations in Stochastic Programming, in: Ermoliev, Y. and Wets, R.J-B, eds., *Numerical Techniques for Stochastic Optimization*, Springer-Verlag, Berlin.

[65] Kall, P. and Wallace, S.W. 1994. *Stochastic Programming.* Wiley, New York.

[66] Karmarkar, U.S. 1979. Convex/Stochastic Programming and Multilocation Inventory Problems. *Naval Research Logistics Quarterly.* **26** 1-19.

[67] Karmarkar, N. 1984. A New Polynomial-Time Algorithm for Linear Programming, *Combinatorica.* **4** 373-395.

[68] King, A. 1987. Stochastic Programming Problems: Examples from the Literature, in: Ermoliev, Y. and Wets, R.J-B., eds., *Numerical Techniques for Stochastic Optimization*, Springer-Verlag, Berlin.

[69] Klein, M. 1966. Markovian Decision Models for Reject Allowance Problems. *Management Science.* **12** 349-358.

[70] Laporte, G., Louveaux, F.V., and van Hamme, L. 1994. Exact Solution of a Stochastic Location Problem by an Integer L-shaped Algorithm. *Transportation Science*, **28** 95-103.

[71] Lee, H.L., and Yano, C.A. 1988. Production Control in Multistage Systems with Variable Yield Losses. *Operations Research.* **36** 269-278.

[72] Lee, H.L., Padmanabhan, V., and Whang, S.J. 1997. Information distortion in a supply chain: The bullwhip effect, *Management Science*, **43** 546-558.

[73] Macario, A., Terry, V.S., Dunn, B., McDonald T., 1995. Where Are the Costs in Perioperative Care?. *Anesthesiology.* **83** 1138-1144.

[74] Mazzola, J.B., McCoy, W.F. and Wagner, H.M. 1987. Algorithms and Heuristics for Variable-Yield Lot Sizing. *Naval Research Logistics.* **34** 67-86.

[75] McGillivray, A. and Silver, E.A. 1978. Some Concepts for Inventory Control Under Substitutable Demand. *INFOR.* **16** 47-63.

[76] Mercer, A., 1960. A Queueing Problem in which the Arrival Times of the Customers are Scheduled. *Journal of the Royal Statistical Society, Series B.* **22** 108-113 .

[77] Mirchandani, P.B. and Francis, R.L. 1990. Discrete Location Theory, Wiley, New York.

[78] Murty, K.G. 1983. Linear Programming, Wiley, New York.

[79] Louveaux, F.V. and Peeters, D. 1992. A Dual-Based Procedure For Stochastic Facility Location. *Operations Research.* **40** 564-573.

[80] Mercer, A. 1973. Queues with Scheduled Arrivals: A Correction Simplification and Extension. *Journal of the Royal Statistical Society, Series B.* **35** 104-116.

[81] Moinzadeh, K.H., and Lee H.L. 1987. A Continuous-Review Inventory Model with Constant Resupply Time and Defective Items. *Naval Research Logistics.* **34** 457-467.

[82] Mulvey, J.M. and Vladimirou, H. 1991. Solving Multistage Stochastic Networks: An Application of Scenario Aggregation. *Networks.* **21** 619-643.

[83] Nemhauser, G.L. and Wolsey, L.A. 1999. Integer and Combinatorial Optimization. Wiley, New York.

[84] Parlar, M. and Goyal, S. 1984. Optimal Ordering Decisions for Two Substitutable Products with Stochastic Demands. *OPSEARCH.* **21** 1-15.

[85] Pereira, M.V.F. and Pinto, L.M.V.G. 1991. Multi-Stage Stochastic Optimization Applied to Energy Planning, *Mathematical Programming.* **52** 359-375.

[86] Pierskala, W.P. and D.J. Brailer 1994. Applications of Operations Research in Health Care Delivery. Ch. 13 in *Operations Research in the Public Sector.* S.M. Pollock, M.H. Rothkopf and A. Barnett (eds.). Vol. 6 of *Handbooks in Operations Research and Management Science.* North-Holland.

[87] Pinter, J. 1991. Stochastic Modelling and Optimization for Environmental Management. *Annals of Operations Research.* **31** 527-544.

[88] Porteus, E.L. 1990. Stochastic Inventory Theory, in: Heyman, D.P. and Sobel, M.J., eds., Handbooks in OR & MS, Vol. 2, Elsevier, North-Holland, 605-652.

[89] Prékopa, A. 1980. Logarithmically Concave Measures and Related Topics, in: Dempster, M.A.H., Ed., *Stochastic Programming*, Academic Press, New York.

[90] Raiffa, H. and Schlaifer, R. 1961. *Applied Statistical Decision Theory*, Harvard University, Boston.

[91] Rao, U.S., Jayashankar, M.S., and Zhang, J. 2000. A Multi-Product Inventory Problem with Setup Costs and Downward Substitution. *Working Paper.* Carnegie Mellon University.

[92] Redelmeier, D.A. and Fuchs, V.R. 1993. Hospital Expenditures in the United States and Canada. *New England Journal of Medicine.* **11** 772-778.

[93] Ridder, A., Van Der Laan, E. and Salomon, M. 1998. How Larger Demand Variability May Lead to Lower Costs in the Newsvendor Problem. *Operations Research.* **6** 934-936.

[94] Robinson, L.W. 1990. Optimal Approximate Policies in Multiperiod, Multilocation Inventory Models With Transshipments. *Operations Research.* **38** 278-295.

[95] Robinson, L.W., Gerchak, Yigal, Gupta, D. 1996. Appointment Times Which Minimize Waiting and Facility Idleness. *Working Paper.* McMaster University.

[96] Robinson, L.W. and Chen R.R. 2000. Scheduling Doctor's Appointments: Optimal and Empirically-Based Heuristic Policies. *Working Paper.* Cornell University.

[97] Rockafellar, R.T. and Wets, R. J-B. 1976. Nonanticipativity and L^1-martingales in Stochastic Optimization Problems, *Mathematical Programming Study.* **6** 170-187.

[98] Rockafellar, R.T. and Wets, R. J-B. 1991. Scenarios and Policy Aggregation in Optimization Under Uncertainty, *Mathematics of Operations Research.* **16** 119-147.

[99] Sabria, F. and Daganzo, C.F. 1989. Approximate Expressions for Queuing Systems with Scheduling Arrivals and Established Service Order. *Transportation Science.* **23** 159-165.

[100] Schultz, R., Stougie, L., and M.H. van der Vlerk. 1996. Two-Stage Stochastic Integer Programming: A Survey. *Statistica Neerlandica.* **50** 404-416.

[101] Sen, A. and Zhang, A.X. 1999. The Newsboy Problem with Multiple Demand Classes. *IIE Transactions.* **31** 431-444 .

[102] Shapiro, A. and Homem-de-Mello, T. 1998. A Simulation-Based Approach to Two-Stage Stochastic Programming with Recourse. *Mathematical Programming.* **81** 301-325.

[103] Shih, W. 1980. Optimal Inventory Policies When Stockouts Result From Defective Products. *International Journal of Production Research* **18** 677-685.

[104] Silver, S.A. and Pyke, D.F. and Peterson, R. 1998. Inventory Management and Production Planning and Scheduling. 3rd Edition. Wiley, New York.

[105] Soriano, A. 1966. Comparison of Two Scheduling Systems. *Operations Research.* **14** 388-397.

[106] Somlyody, L. and Wets, R.J.B. 1988. Stochastic Optimization Models for Lake Eutrophication Management. *Operations Research.* **36** 660-681.

[107] Strazicky, B. 1980. Some Results Concerning an Algorithm for the Discrete Recourse Problem, in: Dempster, M.A.H., Ed., *Stochastic Programming*, Academic Press, New York.

[108] Strum, D.P., Vargas, L.G., Jerrold, M.H. 1999. Surgical Subspecialty Block Utilization and Capacity Planning. *Anesthesiology.* **4** 1176-1185.

[109] Takriti, S. and Birge, J.R. 1995 A Stochastic Model for the Unit Commitment Problem, *IEEE Transactions on Power Systems.* **3** 1497-1508.

[110] Takriti, S. and Birge, J.R. 2000. Lagrangian Solution Techniques and Bounds for Loosely Coupled Mixed-Integer Stochastic Programs, *Operations Research.* **48** 91-98.

[111] Teitz, M.B. and Bart, P. 1967, Heuristic Methods for Estimating the Generalized Vertex Median of a Weighted Graph, *Operations Research*, **16** 955-961.

[112] van der Vlerk, M.H. 2000. Stochastic Integer Programming Bibliography. World Wide Web, http://mally.eco.rug.nl/biblio/stoprog.html.

[113] Van Slyke, R.M. and Wets, R.J-B. 1969. L-shaped Linear Programs with Applications to Optimal Control and Stochastic Programming, *SIAM Journal of Applied Mathematics* . **17** 638-663.

[114] Wald, A. 1950. *Statistical Decision Functions*, Wiley, New York.

[115] Wallace, S.W. 1986. Solving Stochastic Programs With Network Recourse. *Networks.* **16** 295-317.

[116] Welch, J. and Bailey, N. 1952. Appointment Systems in Hospital Outpatient Departments. *The Lancet.* 1105-1108.

[117] Welch, J. 1964. Appointment Systems in Hospital Outpatient Departments. *Operational Research Quarterly.* **15** 224-237.

[118] Weiss, E.N. 1990. Models for Determining Estimated Start Times and Case Orderings in Hospital Operating Rooms. *IIE Transactions.* **22** 143-150.

[119] Wets, R.J-B. 1974. Stochastic Programs with Fixed Recourse: The Equivalent Determini stic Program, *Siam Review.* **16** 309-339.

[120] Wets, R.J-B. 1988. Large-Scale Linear Programming Techniques in Stochastic Programming, in: Ermoliev, Y. and Wets, R.J -B. (eds.): *Numerical Techniques for Stochastic Optimization*, Springer Verlag, Berlin.

[121] White, L.S. 1967. Bayes Markovian Decision Models for a Multiperiod Reject Allowance Problem. *Operations Research.* **15** 857-865.

[122] Williams, A.C. 1962. A Stochastic Transportation Problem. *Operations Research.* **11** 759-770.

[123] White, M. and Pike, M. 1964. Appointment Systems in Outpatient Clinics and the Effect of Patients' Unpunctuality. *Medical Care.* **2** 133-145.

[124] Yano, C.A., Lee, H.L. 1995. Lot Sizing with Random Yields: A Review, *Operations Reserach.* **43** 311-334.

[125] Zipkin, P.H. 1979. Bounds for Row-Aggregation in Linear Programming. *Operations Research.* **28** 903-916.