Stochastic Programs and their Value over Deterministic Programs

by Stuart Corrigall

A dissertation submitted to the Faculty of Arts, University of the Witwatersrand, Johannesburg, in fulfilment of the requirements for the degree of Master of Arts.

Johannesburg, 1998.

71

DECLARATION

I declare that this dissertation is my own, unaided work. It is being submitted for the Degree of Master of Arts at the University of the Witwatersrand, Johannesburg. It has not been submitted before for any degree or examination at any other university.

Stuart Corrigall
February 1998

ABSTRACT

Real-life decision-making problems can often be modelled by mathematical programs (or optimization models). It is common for there to be uncertainty about the parameters of such optimization models. Usually, this uncertainty is ignored and a simplified deterministic program is obtained. Stochastic programs take account of this uncertainty by including a probabilistic description of the uncertain parameters in the model. Stochastic programs are therefore more appropriate or valuable than deterministic programs in many situations, and this is emphasized throughout the dissertation. The dissertation contains a development of the theory of stochastic programming, and a number of illustrative examples are formulated and solved. As a real-life application, a stochastic model for the unit commitment problem facing Eskom (one of the world's largest producers of electricity) is formulated and solved, and the solution is compared with that of the current strategy employed by Eskom.

ACKNOWLEDGEMENTS

I would like to thank Professor Paul Fatti for introducing me to the subject and, as supervisor, for his careful checking and scrutiny of my work and for the many useful comments made. Any errors that remain in the text are solely the responsibility of the author.

Thanks to Dr John Dean of Eskom for providing data and information on Eskom's unit commitment problem, and for obtaining permission to publish the results.

Much of the theory in the dissertation was based on the textbooks by Kall & Wallace [28] and Birge & Louveaux [10]. The examples that appear in the dissertation were either taken directly or adapted from exercises and examples in these textbooks. I would like to thank the authors of these books for helping to make the field of stochastic programming accessible. In particular, I wish to thank Professor John R. Birge of the University of Michigan for permission to publish the example in manufacturing design in Section 4.6 and for confirming my solution to the problem.

I appreciate the financial assistance received from the Foundation for Research and Development (FRD) in the form of a grant and from the University of the Witwatersrand in the form of a Postgraduate Merit Award.

Finally, I would like to thank Professor James Ridley for his assistance with one of the proofs, and to express my gratitude to everybody else who provided assistance of any kind.

CONTENTS

Appendices

LIST OF FIGURES

LIST OF TABLES

Chapter 1

Introduction

This chapter provides an introduction to stochastic programming by describing mathematical programming models and showing how these can be generalized to become stochastic programming models. An overview of the literature on stochastic programming is provided and practical applications of stochastic programming are discussed. The relationship between stochastic programming models and other related models is described, and a simple example of a stochastic program follows. The chapter concludes with a formal discussion of how stochastic programs can be shown to be valuable.

1.1 Mathematical Programming

Many *decision problems* can be modelled by *mathematical programs*, which seek to maximize or minimize some objective function which is a function of the decision variables. The decision variables may be constrained by limits in resources, minimum requirements and maximum requirements. The decision variables may be, for example, non-negative, unrestricted, integral or binary. The objective function and the constraints are functions of the variables and the problem data, or parameters. Examples of parameters are unit costs, production rates, sales and capacities.

1.1.1 Linear Programming Models

Amongst the most common models used are *linear programs* of the form

$$\left.\begin{array}{l} \min\;\; c_1x_1 + c_2x_2 + \ldots + c_nx_n \\ \text{subject to} \\ a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = b_2 \\ \quad\quad\vdots \quad\quad\quad\quad\quad \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n = b_m \\ \quad\quad x_1, x_2, \ldots, x_n \geq 0 \end{array}\right\} \tag{1.1}$$

Using matrix notation, with capital letters for matrices and boldface small letters for vectors, (1.1) becomes

$$\left.\begin{array}{ll} \min & \mathbf{c}^T\mathbf{x} \\ \text{s.t.} & A\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \geq 0 \end{array}\right\} \tag{1.2}$$

where $\mathbf{x} = \begin{pmatrix} x_1 & x_2 & \ldots & x_n \end{pmatrix}^T \qquad \mathbf{c} = \begin{pmatrix} c_1 & c_2 & \ldots & c_n \end{pmatrix}^T$

$$A = \begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \ldots & a_{mn} \end{pmatrix} \quad \mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}$$

Despite the appearance of (1.1) and (1.2), these models do not restrict us to the use of *equality constraints* and *minimization* only. Any *inequality constraint* of the form $\sum_{j=1}^{n} a_{ij}x_j \leq b_i$ can be converted into the equality constraint $\sum_{j=1}^{n} a_{ij}x_j + s_i = b_i$ by adding the non-negative *slack* variable $s_i \geq 0$ to the left-hand side. Similarly, the inequality constraint $\sum_{j=1}^{n} a_{ij}x_j \geq b_i$ can be converted into the equality constraint $\sum_{j=1}^{n} a_{ij}x_j - S_i = b_i$ by subtracting the non-negative *surplus* variable $S_i \geq 0$ from the left-hand side. Models that involve *maximization* of the objective can be converted by rewriting the objective $\max\ \mathbf{c}^T\mathbf{x}$ as $-\min\ \left(-\mathbf{c}^T\mathbf{x}\right)$. Furthermore, it is possible to include *free* variables (*i.e.* variables that are *unrestricted in sign*) in the model. A decision variable y that is unrestricted in sign can be included in the model by rewriting it as the difference $y^+ - y^-$ of the two non-negative decision variables $y^+, y^- \geq 0$. For further background in linear programming, see Appendix A.2.

1.1.2 More General Mathematical Programming Models

In model (1.1) the coefficients c_j , a_{ij} and b_i are assumed to have fixed known real values and we are left with the task of finding an optimal combination of the values for the decision variables x_{ij} that satisfy the given constraints. Model (1.1) can only provide a reasonable representation of a real-life problem when the functions involved are linear in the decision variables, or approximately so. If this condition is *substantially* violated, we should use the following more general form to model the problem

$$\left.\begin{array}{ll} \min & g_0(\mathbf{x}) \\ \text{s.t.} & g_i(\mathbf{x}) \leq 0\,,\ i = 1,\ldots,m \\ & \mathbf{x} \in X \subseteq \mathbb{R}^n \end{array}\right\} \tag{1.3}$$

The form presented in (1.3) is known as a *mathematical programming problem*, and includes both linear and nonlinear programs. The set X as well as the functions $g_i : \mathbb{R}^n \to \mathbb{R}$, $i = 1,\ldots,m$ are given by the modelling process Model (1.3) does not exclude equality constraints, "greater than" constraints, or maximization models. Constraints of the form $g_i(\mathbf{x}) \geq 0$ can be rewritten as $-g_i(\mathbf{x}) \leq 0$, and constraints of the form $g_i(\mathbf{x}) = 0$ can be included as $g_i(\mathbf{x}) \leq 0$ and $-g_i(\mathbf{x}) \leq 0$. The model does not exclude non-zero right-hand sides either, since the constraint $g_i(\mathbf{x}) \leq c$ can be rewritten as $g_i(\mathbf{x}) - c \leq 0$, where c is a constant. The objective $\max g_0(\mathbf{x})$ of a maximization model can be converted to $-\min\left(-g_0(\mathbf{x})\right)$.

Definitions of convex sets, convex functions, convex polyhedral sets and other related concepts are contained in Appendix A.1. Depending on the properties of the functions g_i and the set X, program (1.3) is classified as

1. *Linear* if the set X is convex polyhedral and the functions g_i, $i = 0,\ldots,m$ are linear.
2. *Nonlinear* if at least one of the functions g_i, $i = 0,\ldots,m$ is nonlinear, or X is not a convex

polyhedral set. Amongst nonlinear programs, we classify a program as

(a) *Convex* if the feasible set $\mathcal{B} := X \cap \{\mathbf{x} \mid g_i(\mathbf{x}) \leq 0\,,\ i = 1, \ldots, m\}$ is a convex set and the objective function g_0 is a convex function.

(b) *Nonconvex* if $\mathcal{B}$ is not a convex set or the objective function g_0 is not convex.

For further background in nonlinear programming, see Appendix A.3. *Mixed integer* programs arise if the set X requires some of the variables x_i to take integer values only. A development of integer programming is contained in Appendix A.4. Note that linear programs are convex, but mixed integer programs are not. The following proposition gives a sufficient condition for the feasible set to be convex.

Proposition 1 *If the functions g_i, $i = 1, \ldots, m$ are convex and X is a convex set, then the feasible set $\mathcal{B} = X \cap \{\mathbf{x} \mid g_i(\mathbf{x}) \leq 0\,,\ i = 1, \ldots, m\}$ is a convex set.*

Proof. Let $\mathbf{x}_1, \mathbf{x}_2 \in \{\mathbf{x} \mid g_i(\mathbf{x}) \leq 0\,,\ i = 1, \ldots, m\}$ and let $\bar{\mathbf{x}} = \lambda \mathbf{x}_1 + (1-\lambda)\mathbf{x}_2$ where $\lambda \in (0, 1)$. Then by convexity of g_i, $i = 1, \ldots, m$, it follows that

$$g_i(\bar{\mathbf{x}}) = g_i\left(\lambda \mathbf{x}_1 + (1-\lambda)\mathbf{x}_2\right) \leq \lambda g_i(\mathbf{x}_1) + (1-\lambda) g_i(\mathbf{x}_2) \leq 0$$

for $i = 1, \ldots, m$, since $g_i(\mathbf{x}_1) \leq 0$ and $g_i(\mathbf{x}_2) \leq 0$. Therefore $\bar{\mathbf{x}} \in \{\mathbf{x} \mid g_i(\mathbf{x}) \leq 0\,,\ i = 1, \ldots, m\}$, implying that $\{\mathbf{x} \mid g_i(\mathbf{x}) \leq 0\,,\ i = 1, \ldots, m\}$ is a convex set. It follows immediately that $\mathcal{B}$ is a convex set since it is an intersection of convex sets (see Appendix A.1). ■

Corollary 1 *The constraint $g_i(\mathbf{x}) \geq 0$ defines a convex feasible set if $g_i(\cdot)$ is a concave function of $\mathbf{x}$ and the constraint $g_i(\mathbf{x}) = 0$ defines a convex feasible set if $g_i(\cdot)$ is an affine function of $\mathbf{x}$.*

Proof. The constraint $g_i(\mathbf{x}) \geq 0$ is equivalent to $-g_i(\mathbf{x}) \leq 0$ which defines a convex set if $-g_i$ is a convex function, *i.e.* if g_i is a *concave* function. The constraint $g_i(\mathbf{x}) = 0$ is equivalent to the constraints $g_i(\mathbf{x}) \leq 0$ and $-g_i(\mathbf{x}) \leq 0$ which define a feasible set if both g_i and $-g_i$ are convex functions. This is only true if g_i is an *affine* function. ■

1.1.3 The Importance of Convexity

Definition 1 (Local Minimum) *Program (1.3) is said to attain a relative minimum or local minimum at some point* $\hat{x}$ *if there is a neighbourhood* U *of* $\hat{x}$ *such that* $g_0(\hat{x}) \leq g_0(y)\ \forall y \in U \cap B$.

Definition 2 (Global Minimum) *Program (1.3) is said to attain a global minimum at some point* $\bar{x}$ *if* $g_0(\bar{x}) \leq g_0(y)\ \forall y \in B$.

The following lemma shows that *convex programs* have the useful property that any local minimum will also be a global minimum. We try to *avoid nonconvex programs* whenever possible, since it is difficult in general to find the global minimum of a nonconvex program. This is because many different stationary points (in the form of local minima, local maxima and saddle points) may exist, and when one is found, there is often no easy way to tell whether it is in fact a global minimum. Moreover, for programs with nonconvex feasible sets, one may move out of the feasible set when interpolating between two feasible points. This can be problematic when designing an optimization algorithm.

Lemma 1 *If problem (1.3) is a convex program, then any local minimum is a global minimum.*

Proof. If $\bar{x}$ is a local minimum of problem (1,3), then $\bar{x}$ belongs to the feasible set $B := X \cap \{x \mid g_i(x) \leq 0\,,\ i = 1, \ldots, m\}$. Furthermore, there exists an $\varepsilon_0 > 0$ such that for any ball $K_\varepsilon := \{x \mid \|x - \bar{x}\| \leq \varepsilon\}\,,\ 0 < \varepsilon < \varepsilon_0$, we have that $g_0(\bar{x}) \leq g_0(x)\ \forall x \in K_\varepsilon \cap B$. We now show that $g_0(\bar{x}) \leq g_0(y)$ for any feasible y outside $K_\varepsilon \cap B$, which implies that $\bar{x}$ is a global minimum. Choosing an arbitrary $y \in B$, $y \neq \bar{x}$, we may choose $\varepsilon > 0$ such that $\varepsilon < \|y - \bar{x}\|$ and $\varepsilon < \varepsilon_0$, so that y lies outside K_ε. The line segment $\overline{\bar{x}y}$ intersects the surface of the ball K_ε at a point $\hat{x}$ such that $\hat{x} = \alpha\bar{x} + (1 - \alpha)y$ for some $\alpha \in (0, 1)$. Note that $\hat{x} \in B$ since B is a convex set, and that $g_0(\hat{x}) \leq \alpha g_0(\bar{x}) + (1 - \alpha)g_0(y)$ since the objective g_0 is a convex function. Now $g_0(\bar{x}) \leq g_0(\hat{x})$ since $\hat{x} \in K_\varepsilon \cap B$ and hence $g_0(\bar{x}) \leq \alpha g_0(\bar{x}) + (1 - \alpha)g_0(y)$, which implies that $g_0(\bar{x}) \leq g_0(y)$. ∎

1.1.4 Uncertainty in Mathematical Programs

In many modelling situations it is unreasonable to assume that the coefficients c_j, a_{ij} and b_i in problem (1.1) or the functions g_i in problem (1.3) are deterministically fixed. These coefficients can often be modelled as uncertain parameters that are at best characterized by probability distributions. Thus, depending on the situation that is being modelled, problems (1.1) or (1.3) may not describe appropriately the problem we want to solve. It is in such situations that *stochastic programming* models can be appropriate. Stochastic programming therefore broadens the scope of mathematical modelling.

1.2 What is Stochastic Programming?

The aim of *stochastic programming* is to find optimal decisions in problems that involve uncertain data. The underlying premise of stochastic programming is that the future cannot be perfectly forecast but instead should be considered random or uncertain. In this terminology, *stochastic* is opposed to deterministic, while *programming* refers to the modelling and optimizing of the problem as a linear or nonlinear mathematical program.

Under *deterministic programming*, it is assumed that the parameters of the given problem are known accurately (*i.e.* with certainty). However, for many actual problems, the parameters cannot be known accurately, due to various reasons. The first reason is due to lack of reliable data or simple measurement error. The second and more fundamental reason is that some data represent information about unobserved events or future events (*e.g.* product demand or price in the future) and simply cannot be known with certainty. Stochastic programming aims to take this uncertainty into account.

In some instances, no harm will come from ignoring these uncertainties, and one may rely upon best estimates and parametric sensitivity analysis. However, there are a number of situations where proceeding in this manner produces solutions whose implementation could lead to disaster. For example, designing a production plan without taking into account the inherent uncertainty about future markets may leave the manufacturer exposed to large losses if the evolution of the market does not closely match the predictions.

Stochastic programs are mathematical programs where some of the coefficients or parameters incorporated into the objective function or constraints are uncertain. *Uncertainty* is usually characterized by probability distributions for the parameters, although in practice, the uncertainty can often be modelled more simply as a few scenarios (*i.e.* possible sets of values of the parameters). When uncertainty is modelled into mathematical programs, changes in the decision process occur and new difficulties are brought in. Furthermore, not all problems are solvable. When some of the parameters are random, the solution and optimal objective value of the problem will also be random. A distribution of optimal decisions is generally unimplementable. Ideally, we would like one decision and one optimal objective value.

Several different stochastic models can be built, depending on the type of uncertainty and the time at which decisions must be taken. There is a variety of stochastic programming models in terms of the objective of the decision process, the constraints on the decisions and the relationship to the random elements. Stochastic programs can generally be classified either as recourse problems or as chance-constrained problems. Under both types of problems, we are required to make the de *before* we can observe the outcome of the random parameters. This is known as a *here-and-now* solution, as opposed to a *wait-and-see* solution where we can observe the outcome of the random parameters before making our decision. *Recourse problems* require that we make one decision now and minimize the expected costs of the consequences of that decision. In fact, it is impossible under uncertainty to find a solution that is ideal under all circumstances. This is because the decision taken may turn out, after observation of the random parameters, to be the wrong one (*i.e.* the constraints may be violated, or the cost not minimized). Such violations of the constraints imply *penalty costs* that depend upon the magnitude of the constraint violations. On the other hand, chance-constrained models try to find a decision which ensures that a set of constraints will hold with a certain probability. The *probabilistic constraints* may be required to hold jointly with a certain probability, or separately, each with an associated probability.

1.3 Overview of the Literature

Stochastic programming is not a very well known subject in the operations research community, largely due to the relative scarcity of the literature and the complex nature of the subject. It is mentioned in few textbooks on optimization or operations research, *e.g.* Taha [44], but even then only a very brief introduction is given. Textbooks devoted solely to stochastic programming started coming out in the 1970s, such as those by Vajda [46], Sengupta [41] and Kall [27].

Dantzig's classic reference on linear programming [14] contains a chapter devoted to uncertainty. Unfortunately, these books display vast differences in terminology, notation and theoretical approach. This was recognized by Dempster [18] in his volume on stochastic programming. However, [18] concentrated on the state of the art and is only understandable to those already familiar with the subject.

Finally, Kall & Wallace [28] brought out a basic textbook with the aim of making the subject accessible not only to mathematicians, but also to students and other interested parties who could not or would not approach the field via the journals. Their book contains a theoretical development of stochastic programming as well as algorithms and references. Unfortunately, however, the book contains few simple examples or references to practical applications and implementations.

Recently, Birge & Louveaux [10] published a textbook that provides a wide-ranging first course in stochastic programming suitable for both students and researchers. The book discusses modelling issues, theoretical properties, solution methods and algorithms, approximation techniques and sampling techniques, and contains exercises, worked examples and a case study. All in all, it provides a good introduction to the subject, and many references are provided.

The field of stochastic programming, which is also known as *optimization under uncertainty*, is currently developing rapidly with contributions from many disciplines including operations research, economics, mathematics, probability and statistics. The development of the theory of stochastic programming can be followed through articles that appear in various academic journals devoted to operations research or optimization. These journals also contain many articles devoted to the application of stochastic programming.

1.4 Applications of Stochastic Programming

Practical problems that require stochastic programming are abundant, since whenever a mathematical programming model seems to be plausible for a real life problem, some *randomness* of parameters or functions is likely to occur. In fact, there are few practical decision problems where the modeller is not faced with uncertainty about the values to assign to some of the parameters. For example, in a production problem, the productivities, capacities, inflows, prices, costs, returns, and demands for goods, energy or transportation may be random variables, and these are reflected in the optimization model by random coefficients or parameters. In most practical applications these uncertainties are ignored, even though the consequences of doing so are generally unknown.

One of the first applications of stochastic programming was in airline planning. In the *airline fleet assignment* problem, a decision on the allocation of aircraft to routes is required, subject to unknown demand for seating [21]. A recourse formulation is used, with penalties being incurred for lost passengers.

The *capacity expansion* problem concerns the optimal choices of the timing and levels of investments to meet unknown future demands of a given product. It has been applied in power plant expansion for electricity generation [10].

The *unit commitment* problem is one of the most important problems in electrical power generation. It involves finding an optimal schedule, and a production level, for each generating unit of an electrical utility over a given period of time. The unit commitment decision indicates which generating units are to be in use at each point in time over a scheduling horizon [45]. A South African case study is presented in Chapter 6 where the scheduling horizon is 24 hours and demand is uncertain over this period.

Financial decision-making models are often modelled as stochastic programs and represent one of the largest application areas of stochastic programming. Stochastic programming is applicable to financial decision making since the incorporation of risk into investment decisions is the essence of financial planning. Many references can be found in [49]. Some examples of the application of stochastic programming in finance are:

- Portfolio selection in general insurance [2]
- Project selection and various other economic applications [18]
- Bank asset and liability management [30]
- Management of pension funds through asset-liability modelling [13]

Further applications of stochastic programming have been made in:

- Production planning [20]
- Water resource management [19], [18]
- Forestry planning [22]
- Hospital staffing [29]
- Energy planning [33]
- Natural gas planning [37]
- Governmental policy for carbon dioxide emission [8]

1.5 General Formulation of Stochastic Programs

Random parameters in (1.3) may lead to the problem

$$\left.\begin{array}{ll} \text{"min"} & g_0\left(\mathbf{x},\tilde{\xi}\right) \\ \text{s.t.} & g_i\left(\mathbf{x},\tilde{\xi}\right) \le 0\,,\ i = 1,\ldots,m \\ & \mathbf{x} \in X \subseteq \mathbb{R}^n \end{array}\right\} \tag{1.4}$$

where $\tilde{\xi} = \left(\tilde{\xi}_1,\ldots,\tilde{\xi}_k\right)^T$ is a *random vector* varying over a set $\Xi \subseteq \mathbb{R}^k$. Formally, we assume that a family $\mathcal{F}$ of events (*i.e.* subsets of Ξ) and the probability distribution P on $\mathcal{F}$ are given. Hence for every subset $A \subset \Xi$ that is an event, *i.e.* $A \in \mathcal{F}$, the probability $P(A)$ is known. Furthermore, we assume that the functions $g_i\left(\mathbf{x},\cdot\right) : \Xi \to \mathbb{R}\ \forall \mathbf{x}\,, i$ are random variables themselves, and that the probability distribution P is independent of $\mathbf{x}$. See Appendix B.1 for a background in probability measure theory.

For a particular value of $\mathbf{x}$ in problem (1.4), the constraints may or not be satisfied, and the objective may or may not be minimized, depending on the realization of $\tilde{\xi}$. Problem (1.4) is therefore not a well-defined mathematical program, since the meanings of the objective and the constraints are unclear when we must take a decision on $\mathbf{x}$ before knowing the realization of $\tilde{\xi}$. We must therefore revise the modelling process, thus leading to *deterministic equivalents* for (1.4).

For a given decision $\mathbf{x}$ and realization ξ of $\tilde{\xi}$, the ith constraint of (1.4) is violated if $g_i(\mathbf{x}, \xi) > 0$. For each constraint, we can provide a *recourse* or *second-stage activity* $y_i(\xi)$, that is chosen after observing the realization ξ, to compensate for the constraint's violation (if it is indeed violated), by satisfying $g_i(\mathbf{x}, \xi) - y_i(\xi) \leq 0$. However, such extra activity is assumed to cause an extra cost or *penalty* of q_i per unit. The additional cost, known as the *recourse cost*, amounts to

$$Q(\mathbf{x}, \xi) = \min_{\mathbf{y}} \left\{ \sum_{i=1}^{m} q_i y_i(\xi) \mid y_i(\xi) \geq g_i(\mathbf{x}, \xi), y_i(\xi) \geq 0 \right\} \tag{1.5}$$

The total cost $f_0(\mathbf{x}, \xi)$ for a given realization ξ is made up of the first-stage cost $g_0(\mathbf{x})$, which is independent of ξ, and the recourse (or second-stage) cost $Q(\mathbf{x}, \xi)$, which depends on ξ. Thus

$$f_0(\mathbf{x}, \xi) = g_0(\mathbf{x}) + Q(\mathbf{x}, \xi) \tag{1.6}$$

Suppose that (1.4) models a factory producing m products, with $g_i(\mathbf{x}, \xi)$ being the difference between demand and output of product i. If $g_i(\mathbf{x}, \xi) > 0$ then there is a shortage in product i, relative to the demand. Assuming that the factory is committed to cover the demands, problem (1.5) can be interpreted as buying the shortage y_i of product i in the market, at a price of q_i per unit. This model has exactly one recourse variable for each constraint. The solution to (1.5) is

$$y_i(\xi) = \max(g_i(\mathbf{x}, \xi), 0) \ \forall i \tag{1.7}$$

This situation is known as *simple recourse* since a closed form solution exists for the recourse variables and the recourse function is called *linear* since it is linear in the recourse variables y_i. If we think in more general terms, some or all of the recourse variables could be associated with each constraint. This leads us to the *general recourse program*

$$Q(\mathbf{x}, \xi) = \min_{\mathbf{y} \in Y} \{ q(\mathbf{y}) \mid h_i(\mathbf{y}) \geq g_i(\mathbf{x}, \xi) \ , \ h_i(\mathbf{y}) \geq 0 \ , \ i = 1, \ldots, m \} \tag{1.8}$$

where the recourse vector $\mathbf{y}(\xi) \in Y \subseteq \mathbb{R}^{\overline{n}}$ consists of $\overline{n}$ recourse variables, and $q : \mathbb{R}^{\overline{n}} \to \mathbb{R}$ and $h_i : \mathbb{R}^{\overline{n}} \to \mathbb{R}$ are given functions.

Provided that it is meaningful and acceptable to the decision maker to minimize the expected value of the total costs (*i.e.* first-stage and recourse costs), we consider instead of problem (1.4), its deterministic equivalent, the *two-stage stochastic program with recourse*

$$\min_{x \in X} \left\{ g_0(x) + E_{\tilde{\xi}} \left[Q\left(x, \tilde{\xi}\right) \right] \right\} \tag{1.9}$$

In general, any stochastic program may be written in the form

$$\left.\begin{array}{ll} \min & E_{\tilde{\xi}} \left[f_0 \left(x, \tilde{\xi} \right) \right] \\ \text{s.t.} & E_{\tilde{\xi}} \left[f_i \left(x, \tilde{\xi} \right) \right] \leq 0 \,, \; i = 1, \ldots, m \\ & E_{\tilde{\xi}} \left[f_i \left(x, \tilde{\xi} \right) \right] = 0 \,, \; i = m+1, \ldots, \overline{m} \\ & \quad x \in X \subseteq \mathbb{R}^n \end{array}\right\} \tag{1.10}$$

where the f_i are constructed from the objective and the constraints in (1.4). Formally, the deterministic equivalent (1.10) is a mathematical program. In the two-stage recourse problem (1.9), f_0 represented the total costs and $f_1, \ldots, f_{\overline{m}}$ could be used to describe the first-stage feasible set X. However, the general formulation (1.10) includes other types of deterministic equivalents (such as chance-constrained problems) for the stochastic program (1.4), depending on how the functions f_i are constructed from the problem functions g_j.

1.6 Relationship to Other Decision-Making Models

Although the general form of a stochastic program can apply to virtually all decision-making problems with unknown parameters, there are certain characteristics that typify stochastic programs. Stochastic programs are generalizations of deterministic mathematical programs in which some of the uncontrollable data or parameters are not known with certainty. Some typical features of stochastic programs are:

- Many decision variables with (possibly infinitely) many potential values
- Decisions taken at discrete times
- Expectation functionals used for the objective
- Probability distributions known or partially known

The *relative importance* of these features contrasts with similar areas, such as statistical decision theory, decision analysis, dynamic programming, Markov decision processes, and stochastic control.

The aim of *optimal statistical decision theory* is to determine the best levels of variables that affect the outcome of a random experiment. With the decision variables x in some set X, the random outcomes ω in the experiment's sample space Ω, the associated probability distribution $F(\omega)$, and the reward or loss function $r(x,\omega)$ associated with the experiment under outcome ω and decision x, the basic problem is

$$\max_{x \in X} E_\omega [r(x,\omega) \mid F] = \max_{x \in X} \int_\omega r(x,\omega)\, dF(\omega) \tag{1.11}$$

Problem (1.11) is also the fundamental form of a stochastic program. The major differences between the fields lie in the underlying assumptions about the relative importance of different aspects of the problem.

In *statistical decision theory*, a heavy emphasis is placed on changes in F to some updated distribution $\widehat{F}_{x,\omega}$ that depends on the choice of x and observations of ω. It is implicitly assumed that this part of the analysis dominates any solution procedure. In *stochastic programming*, on the other hand, one assumes that finding the optimal value x and the expectations with known distributions are far more difficult than finding the form of the function r and the changes in the distribution F. The emphasis is on finding a solution after a suitable problem statement has been formulated.

Decision analysis is part of optimal statistical decision theory. The emphasis is on acquiring information about possible outcomes, evaluating the utility associated with various outcomes, and defining a limited set of possible actions, usually in the form of a decision tree.

Dynamic programming and *Markov decision processes* are an important part of stochastic optimization. In these models, optimal actions are sought at generally discrete points in time. The actions are influenced by random outcomes and carry from some state at stage t to another state at stage $t+1$. Low-dimensional, finite spaces in time, state and action are usually identified and a Markovian structure is assumed, so that actions and outcomes depend only on the current state. A backward recursion equation is formed, resulting in an optimal decision for each state at each stage. With large state spaces, this approach can become computationally intractable, although it does form the basis of many algorithms for multistage stochastic programs - see Section 3.5.8. Another approach is to consider an infinite time horizon and to establish a stationary policy so that only an optimal decision for each state is needed for any stage.

Stochastic programming is the study of optimal decision making under uncertainty. The term stochastic programming emphasizes its relation to mathematical programming and algorithmic optimization procedures. These considerations dominate application and research in the field and distinguish stochastic programming from other fields of study.

1.7 Example: The News Vendor Problem

A well-known elementary example in statistical decision theory is the *news vendor problem*. In this section, I show how this problem can be formulated as a two-stage linear stochastic program with simple recourse and show by means of a numerical example how the stochastic model offers an improvement over the deterministic model. The concepts of the value of the stochastic solution (VSS) and the expected value of perfect information (EVPI) are also introduced through the example. These concepts are formally defined in Section 1.8. The problem can be stated as follows.

1.7.1 Statistical Decision Theory Formulation

A news vendor goes to the publisher every morning and buys x newspapers at a price of c per paper. The number of papers available to the vendor is bounded above by the limit u, representing either the news vendor's purchase power or a limit set by the publisher. The vendor then tries to sell as many newspapers as possible at the selling price q, where $q > c$. Any unsold newspapers can be returned to the publisher at a return price r, with $r < c$. The news vendor must decide how many newspapers x to buy every morning. The demand for newspapers varies daily and is described by the random variable $\tilde{\xi}$ with non-negative support $\Xi \subseteq \mathbb{R}_+$. The news vendor cannot return to the publisher during the day to buy more newspapers.

We define y as the number of newspapers sold during the day and w as the number of newspapers returned to the publisher at the end of the day. In statistical decision theory, the solutions y^* of y and w^* of w for a given decision x and realization ξ are usually presented immediately as

$$y^*(\xi) = \min(\xi, x) \tag{1.12}$$

$$w^*(\xi) = \max(x - \xi, 0) \tag{1.13}$$

with the following justification. The news vendor tries to sell as many papers as possible ($\Rightarrow \max y$), but sales can never exceed the demand ($\Rightarrow y \leq \xi$) or the number of newspapers available ($\Rightarrow y \leq x$), so that the number of papers sold is (1.12). The remaining $x - y^* = (1.13)$ papers are returned at the end of the day. Returns only occur when demand is less than the number of newspapers purchased by the vendor (*i.e.* $\xi < x$). For a given decision x, the expected value of the news vendor's *revenue* from sales and returns is $-Q(x)$ where

$$Q(x) = E_{\tilde{\xi}}[-q\min(\xi, x) - r\max(x - \xi, 0)] \tag{1.14}$$

The news vendor aims to maximize profit, or equivalently to *minimize loss*, which equals purchase costs less revenue. The problem is therefore to seek the optimal x in

$$\begin{aligned} \min \quad & cx + Q(x) \\ \text{s.t.} \quad & 0 \leq x \leq u \end{aligned} \tag{1.15}$$

where $Q(x)$ is described by (1.14).

1.7.2 Formulation as a Stochastic Program

The buying decision x must be taken before the demand $\tilde{\xi}$ is observed. The buying decision thus corresponds to a first-stage decision. The number of newspapers sold and returned can then be determined after the decision x has been taken and the demand ξ has been observed. The decisions on y and w are therefore second-stage decisions and depend on the observed demand ξ. For given values of x and ξ, the optimal values of y and w can be formulated as solutions to the linear program

$$Q(x,\xi) = \min \quad -qy(\xi) - rw(\xi)$$
$$\text{s.t.} \quad \begin{cases} y(\xi) & \leq \xi \\ y(\xi) + w(\xi) & \leq x \\ y(\xi) \, , \, w(\xi) & \geq 0 \end{cases} \tag{1.16}$$

Problem (1.16) leads to the solutions (1.12) and (1.13) provided that $q \geq r$ which has already been assumed since $q > c > r$. The news vendor problem can therefore be written in the form of a *two-stage stochastic linear program with simple recourse* as (1.15) where

$$\mathcal{Q}(x) = E_{\tilde{\xi}} \left[Q(x, \tilde{\xi}) \right] \tag{1.17}$$

and $Q(x,\xi)$ is given by (1.16). The problem is a stochastic *linear* program because both the first-stage problem

$$\min_{0 \leq x \leq u} cx$$

and the second-stage problem (1.16) are linear programs. We have *simple recourse* because explicit optimal solutions can be obtained for the second-stage variables, as in (1.12) and (1.13).

Suppose that $\tilde{\xi}$ is a continuous random variable with support on $[0,\infty)$, cumulative distribution function $F(\cdot)$ and probability density function $f(\cdot)$. Then

$$\begin{aligned} \mathcal{Q}(x) &= \int_0^x [-q\xi - r(x-\xi)] \, f(\xi) \, d\xi + \int_x^\infty -qx \, f(\xi) \, d\xi \\ &= -(q-r) \int_0^x \xi \, f(\xi) \, d\xi - qx + (q-r) x F(x) \end{aligned}$$

On integrating by parts, we find that

$$\int_0^x \xi \, f(\xi) \, d\xi = x F(x) - \int_0^x F(\xi) \, d\xi, \text{ provided that } \lim_{\xi \to 0} \xi \, F(\xi) = 0$$

which is satisfied for continuous distributions with non-negative support, since $F(0) = 0 = \lim_{\xi \to 0} F(\xi)$. It follows that

$$\mathcal{Q}(x) = -qx + (q-r) \int_0^x F(\xi) \, d\xi \tag{1.18}$$

The first derivative of $\mathcal{Q}(x)$ is

$$\mathcal{Q}'(x) = -q + (q-r)\,F(x) \tag{1.19}$$

and its second derivative is

$$\mathcal{Q}''(x) = (q-r)\,f(x) \tag{1.20}$$

Now $\mathcal{Q}''(x) \geq 0\ \forall x \in (0,u)$ since $q > r$ and $f(x) \geq 0\ \forall x$ as it is a p.d.f. Therefore $\mathcal{Q}(x)$ is convex and twice continuously differentiable on $[0,u]$, provided that $f(x)$ is continuous on $[0,u]$. In fact, for any two-stage linear stochastic program with recourse, the second-stage function $\mathcal{Q}(\mathrm{x})$ is convex and continuous in x, and is also differentiable when $\widetilde{\boldsymbol{\xi}}$ is a continuous random vector - see Chapter 3.

We wish to minimize $z(x) = cx + \mathcal{Q}(x)$ on the interval $0 \leq x \leq u$. The function $z(x)$ is convex since cx is linear and $\mathcal{Q}(x)$ is convex. Hence the minimum will be attained at $x^* = 0$ if $z'(0) > 0$ $\Rightarrow c - q + (q-r)F(0) > 0$ or at $x^* = u$ if $z'(u) < 0 \Rightarrow c - q + (q-r)F(u) < 0$. Otherwise, the minimum will be at a point x^* satisfying $z'(x^*) = 0 \Rightarrow c - q + (q-r)F(x^*) = 0$. The news vendor's optimal solution is therefore

$$x^* = \left\{ \begin{array}{cc} 0 & \text{if } F(0) > \frac{q-c}{q-r} \\ u & \text{if } F(u) < \frac{q-c}{q-r} \\ F^{-1}\left(\frac{q-c}{q-r}\right) & \text{otherwise} \end{array} \right] \tag{1.21}$$

where $x = F^{-1}(\alpha) \Leftrightarrow \alpha = F(x)$ and the minimal expected loss is

$$z^* = z(x^*) = -(q-c)x^* + (q-r)\int_0^{x^*} F(\xi)\,d\xi \tag{1.22}$$

If the support of $\widetilde{\xi}$ is the interval $[a,b] \subset \mathbb{R}$ where $b \geq u$ then $\mathcal{Q}(x) = -qx + (q-r)\int_a^x F(\xi)\,d\xi$ instead of (1.18) since $F(a) = 0$. The derivatives (1.19) and (1.20) remain unchanged and hence the optimal solution (1.21) remains unchanged. However, (1.20) will have a discontinuity in $(0,u)$ at a if $a \in (0,u)$ and $\lim_{\xi \downarrow a} f(\xi) > 0$, and hence $z(x)$ will no longer be twice continuously differentiable but will remain continuously differentiable.

Numerical Example

Suppose that the cost price is $R1.00$, the selling price is $R2.50$, the return price is $R0.50$, the upper limit on papers is 150, and the demand is uniformly distributed between 50 and 150 papers. Working in cents, we are given $c = 100$, $q = 250$, $r = 50$, $u = 150$ and $\tilde{\xi} \sim U(50, 150)$ with the c.d.f.

$$F(\xi) = \begin{cases} 0 & \text{if } \xi \leq 50 \\ \frac{\xi - 50}{100} & \text{if } 50 \leq \xi \leq 150 \\ 1 & \text{if } \xi \geq 150 \end{cases}$$

Now $\frac{q-c}{q-r} = 0.75$ and $F(0) = 0 < \frac{q-c}{q-r}$ and $F(u) = 1 > \frac{q-c}{q-r}$. We therefore solve $F(x^*) = 0.75$ to obtain the optimal decision $x^* = 125$. For any decision x, the expected loss is given by

$$z(x) = \begin{cases} -15x & \text{if } 0 \leq x \leq 50 \\ 0.1x^2 - 25x + 250 & \text{if } 50 \leq x \leq 150 \end{cases}$$

The news vendor's minimal expected loss is $z(125) = -13125$, corresponding to an expected daily profit of $R131.25$.

1.7.3 Deterministic Formulation

Suppose that, instead of using the above stochastic model with recourse, we simply use a deterministic model where we assume that the demand is known and fixed as $\bar{\xi}$. If the news vendor buys x newspapers at a cost of cx, he will sell $y = \min(x, \bar{\xi})$ and return $w = \max(x - \bar{\xi}, 0)$, resulting in a loss of

$$\begin{aligned} \bar{z}(x) &= cx - qy - rw = cx - q\min(x, \bar{\xi}) - r\max(x - \bar{\xi}, 0) \\ &= \begin{cases} \begin{cases} -(q-c)x & \text{on } 0 \leq x \leq \bar{\xi} \\ (c-r)x - (q-r)\bar{\xi} & \text{on } \bar{\xi} < x \leq u \end{cases} & \text{if } \bar{\xi} < u \\ -(q-c)x \quad \text{on } 0 \leq x \leq u & \text{if } \bar{\xi} \geq u \end{cases} \end{aligned}$$

Under the deterministic model, which is simply a relaxation of the stochastic model and is also known as the *expected value problem*, the loss is minimized by the deterministic solution

$$\overline{x} = \min(\overline{\xi}, u) \tag{1.23}$$

with the minimal loss under the deterministic model being

$$\overline{z}(\overline{x}) = -(q - c)\min(\overline{\xi}, u) \tag{1.24}$$

The news vendor will sell $\min(\overline{\xi}, u)$ papers and return none, an ideal situation.

Coming back to our numerical example, the deterministic solution is $\overline{x} = 100$ and the minimal loss under the deterministic model is $\overline{z}(100) = -(250-100)\times 100 = 15000$, corresponding to a profit of $R150.00$, which is greater than that of the stochastic model! The deterministic model *seems* to produce a better objective value (*i.e.* higher profit) than the stochastic model. This, however, is misleading since the deterministic model ignores the uncertainty that is inherent in the problem. Under the stochastic model, we aimed to minimize the expected loss $z(x)$ (as shown in Figure 1-1) of taking the decision x, which resulted in the optimal decision $x^* = 125$ and the minimal expected loss $z(x^*) = -13125$. If we use the deterministic solution $\overline{x} = 100$ in the presence of uncertainty that actually exists, the expected loss will be $z(\overline{x}) = -12500$ which is greater (*i.e.* worse) than the expected loss that arises when the stochastic solution is used. Clearly the stochastic solution will be better in the long run than the deterministic solution. The amount by which it is better is known as the *value of the stochastic solution* or VSS, and in this case is $-12500 - (-13125) = 625$, corresponding to an increase in profit for the news vendor of $R6.25$ per day. In percentage terms, the stochastic solution offers an improvement of $625/12500 = 5\%$ over the deterministic solution. Figure 1-1 shows the graph of the expected loss that we are seeking to minimize and the difference VSS in expected loss between using the stochastic solution and the deterministic solution.

1.7.4 Perfect Information

Suppose that we knew in advance what the demand for the day would turn out to be. This situation is known as having *perfect information*. The problem would then be exactly as the

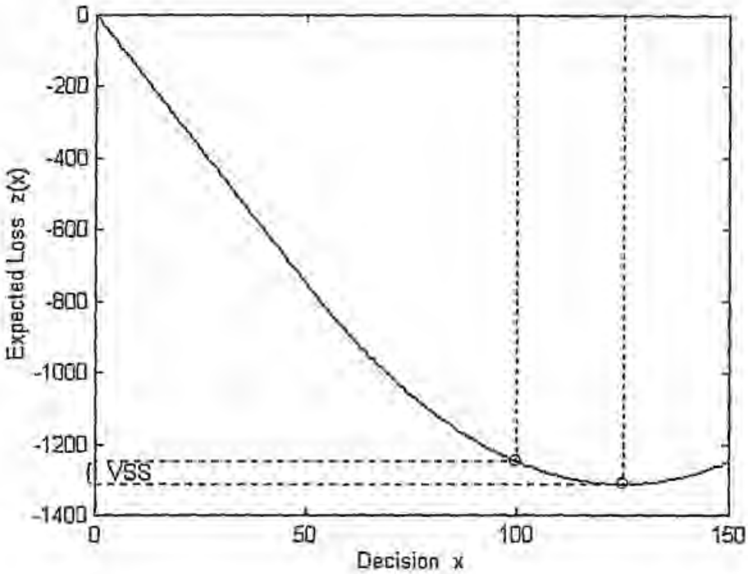

Figure 1-1: The Value of the Stochastic Solution

deterministic formulation, with $\bar{\xi}$ replaced by ξ. Our optimal decision for a given ξ would be to buy and sell $\min(\xi, u)$ papers and to return none, yielding a loss of $(c-q)\min(\xi, u)$. Thus the average loss under perfect information is

$$\int_0^\infty (c-q)\min(\xi, u)f(\xi)d\xi \tag{1.25}$$

which in our example amounts to

$$\int_{50}^{150} (100-250)\xi(0.01)\,d\xi = -15000$$

corresponding to an expected profit of $R150.00$, which is better than expected profit in the situation where we are only given distributional information on the demand. The difference between the expected loss under perfect information and the expected loss when we are only given distributional information (and take the stochastic solution) is known as the *expected* *value of perfect information* or EVPI. In our example, $EVPI = -13125 - (-15000) = 1875$, corresponding to an expected increase in daily profit of $R18.75$, or $1875/13125 = 14.3\%$. In reality, however, it is not possible for the news vendor to have perfect information about the demand, and therefore perfect information is a purely theoretical concept. The best that the vendor can do in *reality* is to take the stochastic solution.

The average decision taken under perfect information is

$$E_{\tilde{\xi}}[x(\xi)] = E_{\tilde{\xi}}[\min(\xi, u)] = \int_{50}^{150} 0.01\xi \, d\xi = 100$$

which in this case is the same as the deterministic solution $\bar{x}$. In this example, if we take this decision under uncertainty instead of the stochastic solution, we will be no better off than if we take the deterministic solution. I call this solution as the *average wait-and-see solution*.

1.8 The Value of Stochastic Programming

1.8.1 Alternatives to Stochastic Programming

Stochastic programs are reputedly computationally difficult to solve. The result of this is that when faced with a real-world problem, many people are inclined to a solve *simpler versions* of the problem, such as the deterministic relaxation obtained by replacing all the random variables by their expected values. Another commonly used approach is to solve several deterministic programs, each of which corresponds to a particular scenario, and then to combine these different solutions by some heuristic rule.

An alternative approach is to assume *pessimistic* (*i.e.* unfavourable) values for all of the random variables in the model, a process known as the inclusion of *fat* or *slack* in the model - see Chapter 25 of Dantzig [14]. The solution that results is a *conservative* solution called a *fat solution*. It is hoped that the fat will act as a *shock absorber* that will prevent recourse actions having to be taken for nearly all possible realizations of the random variables. Fat solutions tend to have high first-stage costs but very small second-stage costs - see Sections 3.7.3 and 3.7.11 for an example of a fat solution that illustrates this point.

The question arises as to whether these approaches are nearly optimal or whether they are totally inaccurate. The value of the stochastic solution (VSS) and the expected value of perfect information (EVPI) provide an answer to this question.

1.8.2 The Expected Value of Perfect Information

The *expected value of perfect information*, or EVPI, measures the maximum amount that a decision maker would be prepared to pay in return for complete and accurate information about the future. The concept of EVPI is often presented in the context of Bayesian decision analysis in statistical decision theory. In the setting of stochastic programming, we define EVPI as follows. Suppose that the uncertainty can be modelled by a number of scenarios, corresponding to the various possible realizations of the random vector $\tilde{\xi}$; and that the objective $z(\mathbf{x},\xi)$ of the problem is to be minimized and depends on both the realization ξ and the decision vector $\mathbf{x} \in X \subseteq \mathbb{R}^n$.

This section provides a number of definitions of concepts that are used to show the value of stochastic programming and perfect information. In the literature, these definitions do not appear to be completely standardized and confusion can result. This is particularly true for the term "wait-and-see solution", which in Kall & Wallace [28] is used to describe a solution of the wait-and-see problem, while in Birge & Louveaux [10], it is used to describe the expected optimal objective value when the wait-and-see approach is used. Although the former definition makes sense, the latter definition appears to be illogical since the expected objective is not a solution at all. Due to this disparity in the literature, the following section contains precise definitions of these concepts, even though some of the definitions may seem redundant.

Definition 3 (Wait-and-See Problem) *The wait-and-see problem for a particular realization ξ of $\tilde{\xi}$ is the mathematical program*

$$\min_{\mathbf{x}\in X} z(\mathbf{x},\xi) \tag{1.26}$$

Definition 4 (Wait-and-See Solution) *Let $\overline{\mathbf{x}}(\xi)$ be an optimal solution to the wait-and-see problem (1.26) for a given realization ξ. It is known as the wait-and-see solution for realization ξ and satisfies*

$$z\left(\overline{\mathbf{x}}(\xi),\xi\right) = \min_{\mathbf{x}\in X} z(\mathbf{x},\xi) \tag{1.27}$$

In a scenario approach, one might be interested in finding the optimal solution $\overline{\mathrm{x}}(\xi)\ \forall\xi$ (*i.e.* for all scenarios) and the associated optimal objective values $z\left(\overline{\mathrm{x}}(\xi),\xi\right)\ \forall\xi$. This is known as the *distribution problem* since it involves finding the distributions of the random solution vector $\overline{\mathrm{x}}(\tilde{\xi})$ and the random objective $z\left(\overline{\mathrm{x}}(\tilde{\xi}),\tilde{\xi}\right)$.

Definition 5 (WS) *The expected value of the optimal objective function of the wait-and-see problem (1.26) can be called the expected wait-and-see objective, is denoted WS and is defined as*

$$WS = E_{\tilde{\xi}}\left[\min_{\mathrm{x}\in X} z(\mathrm{x},\tilde{\xi})\right] = E_{\tilde{\xi}}\left[z\left(\overline{\mathrm{x}}(\tilde{\xi}),\tilde{\xi}\right)\right] \tag{1.28}$$

I propose a solution that I call the *average wait-and-see solution* and show by means of examples in this text that this solution is not only difficult to calculate, but is often even worse than the deterministic solution.

Definition 6 (Average Wait-and-See Solution) *I define the average wait-and-see solution as $E_{\tilde{\xi}}\left[\overline{\mathrm{x}}(\tilde{\xi})\right]$ and the expected result of using the average wait-and-see solution as $AWS = E_{\tilde{\xi}}\left[z\left(E_{\tilde{\xi}}\left[\overline{\mathrm{x}}(\tilde{\xi})\right],\tilde{\xi}\right)\right]$.*

A *here-and-now solution* is one that must be taken before the realization of a random vector $\tilde{\xi}$. This is opposed to a wait-and-see solution where the decision can be taken after $\tilde{\xi}$ has been realized. The optimal here-and-now solution corresponds to the solution of the recourse problem. The essential difference between the wait-and-see approach and the here-and-now approach is in the order of operations. Under the wait-and-see approach, one first optimizes and then takes expectations, while in the here-and-now approach, one first takes expectations and then optimizes.

Definition 7 (Recourse Problem) *The recourse problem is the mathematical program*

$$\min_{\mathrm{x}\in X} E_{\tilde{\xi}}\left[z\left(\mathrm{x},\tilde{\xi}\right)\right] \tag{1.29}$$

Definition 8 (Recourse Solution) *The optimal solution x^* of the recourse problem (1.29) is called the optimal here-and-now solution, the stochastic solution or the recourse solution and satisfies*

$$E_{\tilde{\xi}}\left[z\left(\mathrm{x}^*,\tilde{\xi}\right)\right] = \min_{\mathrm{x}\in X} E_{\tilde{\xi}}\left[z\left(\mathrm{x},\tilde{\xi}\right)\right] \tag{1.30}$$

Definition 9 (RP) *The optimal objective value of the recourse problem, which can also be called the expected here-and-now objective, is denoted RP and is defined as*

$$RP = \min_{x \in X} E_{\tilde{\xi}} \left[z\left(x, \tilde{\xi}\right) \right] = E_{\tilde{\xi}} \left[z\left(x^*, \tilde{\xi}\right) \right] \tag{1.31}$$

Definition 10 (EVPI) *The expected value of perfect information or EVPI is defined as the expected difference between the here-and-now objective and the wait-and-see objective, i.e.*

$$EVPI = RP - WS \tag{1.32}$$

EVPI measures the extent of *suboptimality* caused by taking the here-and-now solution as opposed to a wait-and-see solution. In reality, however, the best solution to take is often the here-and-now solution, since waiting and seeing may not be possible, or perfect information may not be available at any price.

1.8.3 The Value of the Stochastic Solution

The wait-and-see approach delivers a set of solutions instead of one solution that would be implementable. Such a set of solutions is not only unimplementable, but requires much computational effort. It is tempting to solve a much simpler problem: the one obtained by replacing all the random variables by their expected values. This problem is known as the *expected value problem*, mean value problem or *deterministic relaxation*.

Definition 11 (Expected Value Problem) *If we denote the expectation of the random vector $\tilde{\xi}$ by $\bar{\xi}$, the expected value problem is the mathematical program*

$$\min_{x \in X} z\left(x, \bar{\xi}\right) \tag{1.33}$$

Definition 12 (EV) *The optimal objective of the expected value problem (1.33) is denoted EV and is defined by*

$$EV = \min_{x \in X} z\left(x, \bar{\xi}\right) \tag{1.34}$$

Definition 13 (Expected Value Solution) *The optimal solution $\overline{x}(\overline{\xi})$ to the expected value problem (1.33) is called the expected value solution or deterministic solution and satisfies*

$$z\left(\overline{x}(\overline{\xi}), \overline{\xi}\right) = \min_{x \in X} z\left(x, \overline{\xi}\right) \tag{1.35}$$

Definition 14 (EEV) *The expected result of using the deterministic solution is denoted EEV and is defined by*

$$EEV = E_{\tilde{\xi}}\left[z\left(\overline{x}(\overline{\xi}), \tilde{\xi}\right)\right] \tag{1.36}$$

If there is uncertainty inherent in the problem, it could be unwise to implement the deterministic solution $\overline{x}(\overline{\xi})$ instead of the stochastic solution x^*, as there is no guarantee that $\overline{x}(\overline{\xi})$ will be close to x^* or, more importantly, that EEV will be close to RP. The quantity EEV measures how the deterministic solution performs, by fixing the first-stage decisions as $\overline{x}(\overline{\xi})$ and allowing the second-stage decisions to be chosen optimally as functions of $\overline{x}(\overline{\xi})$ and ξ for each possible realization of $\tilde{\xi}$.

Definition 15 (VSS) *The value of the stochastic solution or VSS can be defined as the additional expected loss that arises from using the deterministic solution (i.e. the expected value solution) as opposed to the stochastic solution (i.e. the recourse solution).*

$$VSS = EEV - RP \tag{1.37}$$

VSS is the *cost of ignoring uncertainty* in choosing a decision. It measures how good or how bad the deterministic decision $\overline{x}(\overline{\xi})$ is under the recourse model (1.29). An alternative interpretation is that VSS measures the extent of suboptimality caused by taking the deterministic decision as opposed to the stochastic decision.

1.8.4 Properties of VSS and EVPI

Proposition 2 *For any stochastic program,*

$$WS \leq RP \leq EEV \tag{1.38}$$

$$EVPI \geq 0 \tag{1.39}$$

$$SS \geq 0 \tag{1.40}$$

Proof. $z(\overline{\mathbf{x}}(\xi), \xi) \leq z(\mathbf{x}^*, \xi)$ $\forall \xi$ since $\overline{\mathbf{x}}(\xi)$ minimizes (1.26). Taking expectations of both sides yields $E_{\tilde{\xi}}\left[z\left(\overline{\mathbf{x}}(\tilde{\xi}), \tilde{\xi}\right)\right] \leq E_{\tilde{\xi}}\left[z\left(\mathbf{x}^*, \tilde{\xi}\right)\right] \Leftrightarrow WS \leq RP$ which proves the first inequality of (1.38). It follows directly that $RP - WS \geq 0 \Leftrightarrow EVPI \geq 0$ by (1.32), thus proving (1.39). Since $\mathbf{x}^*$ minimizes (1.29), it follows that $E_{\tilde{\xi}}\left[z\left(\mathbf{x}^*, \tilde{\xi}\right)\right] \leq E_{\tilde{\xi}}\left[z\left(\overline{\mathbf{x}}(\overline{\xi}), \tilde{\xi}\right)\right] \Leftrightarrow RP \leq EEV$ which proves the second inequality of (1.38). Once again, it follows directly that $EEV - RP \geq 0 \Leftrightarrow VSS \geq 0$ by (1.37), thus proving (1.40). ■

The proposition shows that EVPI and VSS are both non-negative, as would be reasonably expected. Examples exist where $EVPI = 0$ and $VSS > 0$ and other examples exist where $EVPI > 0$ and $VSS = 0$. See Section 4.4 in Birge & Louveaux [10] for such examples. There is no simple rule to relate the two quantities. It would be very useful to know when EVPI and VSS take on large values and when they take on small values, as we would then know precisely when stochastic programming should or shouldn't be used. Only [illegible]tic programs with a large VSS would require the solution of a stochastic program, while [illegible] programs with a large EVPI, it would be worthwhile to find closer-to-perfect information on the uncertainty, if possible.

Intuitively, one feels that using stochastic programming is more relevant when there is *more randomness* in the problem. For example, one might expect that for a given problem, EVPI and VSS would increase when the variances of the random variables increase. However, this is not always true. I conjecture that stochastic programming is likely to be relevant when some of the important parameters in the problem have large coefficients of variation, *i.e.* the uncertainty in the parameters is large relative to their estimated or expected values.

Chapter 2

Probabilistic Programming

In Vajda [46], the terms *probabilistic programming* and *stochastic programming* are used interchangeably. However, Birge & Louveaux [10] use the term *probabilistic programming* to refer to stochastic programs where the description of recourse variables is avoided through modelling with chance constraints or probabilistic objectives. The latter definition is used in this chapter. Nevertheless, the distinction is not entirely clear since models with chance constraints and recourse variables can be devised, as in the example at the end of the chapter.

The chapter starts with the derivation and properties of probabilistic programs in general. Linear chance-constrained problems are examined in detail, along with an example. A discussion of solution methods follows and the chapter concludes with an illustrative example in water resource management, in which alternative possible models are formulated, solved and compared under different distributional assumptions.

2.1 Models in Probabilistic Programming

In probabilistic programming, some of the constraints or the objective are expressed as probabilistic statements about first-stage decisions. The description of second-stage or recourse actions is thus avoided. This is particularly useful when the costs or benefits of second-stage decisions are *difficult to quantify*. Most of the results in probabilistic programming concentrate on converting the probabilistic problem into an equivalent deterministic problem, and on the convexity properties of these deterministic equivalents.

The objective is usually an expectation (the *E-model*), but it may also be the probability of some occurrence (the *P-model*) or even the variance of some result (the *V-model*). Most probabilistic programs are *chance-constrained programs* that optimize an expected value subject to *chance constraints*, which are often called *probabilistic constraints*. Chance constraints are constraints that do not need to hold with certainty (as is always the case in deterministic mathematical programming) or almost surely, but instead must hold with some specified probability level, known as the *reliability level*.

2.2 Derivation of Probabilistic Programs

In Section 1.5 we stated that any stochastic program could be written in the form (1.10). In this section we show how probabilistic programs with joint chance constraints or separate chance constraints, and problems with probabilistic objectives fit into the general stochastic programming formulation (1.10).

2.2.1 Joint Chance Constraints

In terms of (1.10) we define

$$f_0(\mathrm{x},\xi) \quad := g_0(\mathrm{x},\xi) \tag{2.1a}$$

$$f_1(\mathrm{x},\xi) \quad := \left\{ \begin{array}{ll} \alpha - 1 & \text{if } g_i(\mathrm{x},\xi) \le 0,\ \forall i = 1,\ldots,m \\ \alpha & \text{otherwise} \end{array} \right] \tag{2.1b}$$

where $\alpha \in [0,1]$. Now, with the vector-valued function $\mathrm{g}(\mathrm{x},\xi) = \left[\begin{array}{ccc} g_1(\mathrm{x},\xi) & \ldots & g_m(\mathrm{x},\xi) \end{array}\right]^T$,

$$\begin{aligned} E_{\tilde{\xi}}\left[f_1(\mathrm{x},\tilde{\xi})\right] &= \int_{\Xi} f_1(\mathrm{x},\xi)dP(\xi) = \int_{\{\mathrm{g}(\mathrm{x},\xi)\le 0\}} (\alpha-1)dP(\xi) + \int_{\{\mathrm{g}(\mathrm{x},\xi)\not\le 0\}} \alpha dP(\xi) \\ &= (\alpha-1)\Pr\left[\mathrm{g}(\mathrm{x},\tilde{\xi}) \le 0\right] + \alpha \Pr\left[\mathrm{g}(\mathrm{x},\tilde{\xi}) \not\le 0\right] = \alpha - \Pr\left[\mathrm{g}(\mathrm{x},\tilde{\xi}) \le 0\right] \end{aligned}$$

Therefore

$$E_{\tilde{\xi}}\left[f_1(\mathrm{x},\tilde{\xi})\right] \le 0 \Rightarrow \alpha - \Pr\left[\mathrm{g}(\mathrm{x},\tilde{\xi}) \le 0\right] \le 0$$

and hence (1.10) becomes the stochastic program with *joint chance constraints*

$$\left.\begin{array}{ll}\min & E_{\tilde{\xi}}\left[g_0\left(\mathrm{x},\tilde{\xi}\right)\right] \\ \text{s.t.} & \Pr\left[g_i(\mathrm{x},\tilde{\xi}) \leq 0\,,\ \forall i=1,\ldots,m\right] \geq \alpha \\ & \mathrm{x} \in X\end{array}\right\} \tag{2.2}$$

2.2.2 Separate Chance Constraints

In an analogous way to the above, we define in terms of (1.10),

$$f_0(\mathrm{x},\xi) \ :=g_0(\mathrm{x},\xi) \tag{2.3a}$$

$$f_i(\mathrm{x},\xi) \ := \left\{\begin{array}{ll}\alpha_i-1 & \text{if } g_i(\mathrm{x},\xi)\leq 0 \\ \alpha_i & \text{otherwise}\end{array}\right] \text{ for } i=1,\ldots,m \tag{2.3b}$$

where $\alpha_i \in [0,1]\ \forall i$. Now for $i=1,\ldots,m$

$$\begin{aligned} E_{\tilde{\xi}}\left[f_i(\mathrm{x},\tilde{\xi})\right] &= \int_{\Xi} f_i(\mathrm{x},\xi)dP(\xi) = \int_{\{g_i(\mathrm{x},\xi)\leq 0\}}(\alpha_i-1)dP(\xi) + \int_{\{g_i(\mathrm{x},\xi)>0\}}\alpha_i\,dP(\xi) \\ &= (\alpha_i-1)\Pr\left[g_i(\mathrm{x},\tilde{\xi})\leq 0\right] + \alpha_i\Pr\left[g_i(\mathrm{x},\tilde{\xi})>0\right] = \alpha_i - \Pr\left[g_i(\mathrm{x},\tilde{\xi})\leq 0\right] \end{aligned}$$

Therefore

$$E_{\tilde{\xi}}\left[f_i(\mathrm{x},\tilde{\xi})\right] \leq 0\ \forall i \Rightarrow \alpha_i - \Pr\left[g_i(\mathrm{x},\tilde{\xi})\leq 0\right] \leq 0\ \forall i$$

and hence (1.10) becomes the stochastic program with *separate chance constraints*

$$\left.\begin{array}{ll}\min & E_{\tilde{\xi}}\left[g_0\left(\mathrm{x},\tilde{\xi}\right)\right] \\ \text{s.t.} & \Pr\left[g_i(\mathrm{x},\tilde{\xi}) \leq 0\right] \geq \alpha_i\,,\ \forall i=1,\ldots,m \\ & \mathrm{x} \in X\end{array}\right\} \tag{2.4}$$

2.2.3 Probabilistic Objectives

It is also possible to use indicator functions to express a probabilistic objective in a form that is consistent with (1.10). Consider an event $A(x) \subset \Xi$ that depends on x. Let

$$f_0(x,\xi) = \left\{ \begin{array}{ll} 1, & \text{if } \xi \in A(x) \\ 0, & \text{if } \xi \notin A(x) \end{array} \right] \tag{2.5}$$

The objective of (1.10) then becomes

$$\begin{aligned} \min E_{\tilde{\xi}} \left[f_0(\mathrm{x}, \tilde{\xi}) \right] &= \min \int_{\Xi} f_0(\mathrm{x}, \xi) dP(\xi) \\ &= \min \left\{ \int_{\xi \in A(x)} 1\, dP(\xi) + \int_{\xi \notin A(x)} 0\, dP(\xi) \right\} \\ &= \min P\left(\xi \mid \xi \in A(x)\right) = \min P\left(A(x)\right) \end{aligned}$$

Such an objective has the form of a P-model.

2.3 General Properties of Chance Constraints

In this section, which is based on Section 1.5 of Kall & Wallace [28], we show that while chance constraints do not in general define convex feasible sets, it is possible under certain conditions to assert convexity and closedness of the feasible region.

2.3.1 Possible Nonconvexity

Consider the joint probabilistic constraints of the form

$$\Pr\left[\mathrm{g}(\mathrm{x}, \tilde{\xi}) \leq 0 \right] = P\left(\{ \xi \mid \mathrm{g}(\mathrm{x}, \xi) \leq 0 \} \right) \geq \alpha \tag{2.6}$$

A point $\hat{\mathrm{x}}$ is feasible iff the set

$$S(\hat{\mathrm{x}}) = \{ \xi \mid \mathrm{g}(\hat{\mathrm{x}}, \xi) \leq 0 \} \tag{2.7}$$

has a probability measure $P(S(\widehat{\mathbf{x}}))$ of at least α. Alternatively, if $\mathcal{G} \subset \mathcal{F}$ is the collection of all events of $\mathcal{F}$ such that $P(G) \geq \alpha \; \forall G \in \mathcal{G}$ then $\widehat{\mathbf{x}}$ is feasible iff we can find at least one event $\widetilde{G} \in \mathcal{G}$ such that $\mathbf{g}(\widehat{\mathbf{x}}, \boldsymbol{\xi}) \leq 0 \; \forall \boldsymbol{\xi} \in \widetilde{G}$. Formally, $\widehat{\mathbf{x}}$ is feasible iff there exists $G \in \mathcal{G}$ such that

$$\widehat{\mathbf{x}} \in \bigcap_{\xi \in G} \{\mathbf{x} \mid \mathbf{g}(\mathbf{x}, \xi) \leq 0\} \tag{2.8}$$

Now because the general point $\mathbf{x}$ is feasible iff (2.6) is true, it follows that the feasible set $\mathcal{B}(\alpha)$ for a given level of α can be written as

$$\begin{aligned} \mathcal{B}(\alpha) &= \{\mathbf{x} \mid P(\{\boldsymbol{\xi} \mid \mathbf{g}(\mathbf{x}, \boldsymbol{\xi}) \leq 0\}) \geq \alpha\} & (2.9) \\ &= \bigcup_{G \in \mathcal{G}} \bigcap_{\xi \in G} \{\mathbf{x} \mid \mathbf{g}(\mathbf{x}, \xi) \leq 0\} & (2.10) \end{aligned}$$

since the feasible set $\mathcal{B}(\alpha)$ is the union of all vectors $\mathbf{x}$ that are feasible according to (2.8). Although an intersection of convex sets is convex, a union of convex sets is not necessarily convex. Hence we cannot, in general, expect $\mathcal{B}(\alpha)$ to be convex, even if the sets $\{\mathbf{x} \mid \mathbf{g}(\mathbf{x}, \boldsymbol{\xi}) \leq 0\}$ are convex $\forall \boldsymbol{\xi} \in \Xi$. In fact, simple examples exist where chance-constrained problems define nonconvex feasible sets - see Section 1.5 in K ll & Wallace [28] for such an example.

2.3.2 Conditions for Convexity

Although chance constraints can easily define nonconvex feasible sets, it is possible to assert convexity of the feasible region $\mathcal{B}(\alpha)$ for general α under appropriate assumptions on $\mathbf{g}$ and the probability measure P.

Definition 16 (Quasi-Concave Probability Measure) *The probability measure P is quasi-concave if for any convex sets $S_1, S_2 \in \mathcal{F}$,*

$$P(\lambda S_1 + (1-\lambda) S_2) \geq \min[P(S_1), P(S_2)] \;\; \forall \lambda \in [0, 1] \tag{2.11}$$

where $\lambda S_1 + (1-\lambda)S_2 := \{\boldsymbol{\xi} := \lambda \boldsymbol{\xi}^1 + (1-\lambda)\boldsymbol{\xi}^2 \mid \boldsymbol{\xi}^1 \in S_1, \boldsymbol{\xi}^2 \in S_2\}$.

Proposition 3 *If $\mathbf{g}(\cdot, \cdot)$ is jointly convex in $(\mathbf{x}, \boldsymbol{\xi})$ and P is quasi-concave, then the feasible set $\mathcal{B}(\alpha) = \{\mathbf{x} \mid P(\{\boldsymbol{\xi} \mid \mathbf{g}(\mathbf{x}, \boldsymbol{\xi}) \leq 0\}) \geq \alpha\}$ is convex $\forall \alpha \in [0, 1]$.*

Proof. Define $S(\mathbf{x}) := \{\boldsymbol{\xi} \mid \mathbf{g}(\mathbf{x}, \boldsymbol{\xi}) \leq 0\}$, so that $B(\alpha) = \{\mathbf{x} \mid P(S(\mathbf{x})) \geq \alpha\}$. We first show that $S(\mathbf{x})$ is a convex set and then show that $B(\alpha)$ is a convex set. Let $\boldsymbol{\xi}^{(1)}, \boldsymbol{\xi}^{(2)} \in S(\mathbf{x})$. Then

$$\mathbf{g}(\mathbf{x}, \lambda\boldsymbol{\xi}^{(1)} + (1-\lambda)\boldsymbol{\xi}^{(2)}) \leq \lambda\, \mathbf{g}(\mathbf{x}, \boldsymbol{\xi}^{(1)}) + (1-\lambda)\mathbf{g}(\mathbf{x}, \boldsymbol{\xi}^{(2)}) \leq 0$$

since $\mathbf{g}(\mathbf{x}, \cdot)$ is convex in $\boldsymbol{\xi}$ by assumption. This implies that

$$\lambda\boldsymbol{\xi}^{(1)} + (1-\lambda)\boldsymbol{\xi}^{(2)} \in S(\mathbf{x})$$

and therefore $S(\mathbf{x})$ is a convex set.

Let $\mathbf{x}^1, \mathbf{x}^2 \in B(\alpha)$, $\boldsymbol{\xi}^1 \in S(\mathbf{x}^1)$, $\boldsymbol{\xi}^2 \in S(\mathbf{x}^2)$ and $\lambda \in [0,1]$. Then for $\overline{\mathbf{x}} = \lambda\mathbf{x}^1 + (1-\lambda)\mathbf{x}^2$ and $\overline{\boldsymbol{\xi}} = \lambda\boldsymbol{\xi}^1 + (1-\lambda)\boldsymbol{\xi}^2$, it follows by the joint convexity of g in $(\mathbf{x}, \boldsymbol{\xi})$ that

$$\mathbf{g}(\overline{\mathbf{x}}, \overline{\boldsymbol{\xi}}) \leq \lambda\mathbf{g}(\mathbf{x}^1, \boldsymbol{\xi}^1) + (1-\lambda)\mathbf{g}(\mathbf{x}^2, \boldsymbol{\xi}^2) \leq 0$$

Therefore $\overline{\boldsymbol{\xi}} \in S(\overline{\mathbf{x}})$ and hence

$$S(\overline{\mathbf{x}}) \supset \lambda S(\mathbf{x}^1) + (1-\lambda)S(\mathbf{x}^2)$$

since every element of $\lambda S(\mathbf{x}^1) + (1-\lambda)S(\mathbf{x}^2)$ is an element of $S(\overline{\mathbf{x}})$. Therefore

$$\begin{aligned} P(S(\overline{\mathbf{x}})) &\geq P\left(\lambda S(\mathbf{x}^1) + (1-\lambda)S(\mathbf{x}^2)\right) \\ &\geq \min\left[P\left(S(\mathbf{x}^1)\right), P\left(S(\mathbf{x}^2)\right)\right] \geq \alpha \end{aligned}$$

since P is a quasi-concave probability measure and the sets $S(\mathbf{x}^1)$ and $S(\mathbf{x}^2)$ are convex, as shown above. Also $\mathbf{x}^1, \mathbf{x}^2 \in B(\alpha) \Rightarrow P\left(S(\mathbf{x}^1)\right) \geq \alpha$ & $P\left(S(\mathbf{x}^2)\right) \geq \alpha$. It follows that $\overline{\mathbf{x}} \in B(\alpha)$ and therefore $B(\alpha)$ is a convex set $\forall \alpha \in [0,1]$. ■

We have shown that for the class of quasi-concave probability measures, the feasible set is convex for appropriate g. Next, we introduce the class of log-concave probability measures, and show that it is a subclass of quasi-concave probability measures.

Definition 17 (Log-Concave Probability Measure) *The probability measure P is log-concave if for any convex sets* $S_1, S_2 \in \mathcal{F}$ *it satisfies*

$$P(\lambda S_1 + (1-\lambda)S_2) \geq P^{\lambda}(S_1)P^{1-\lambda}(S_2) \; \forall \lambda \in [0,1] \tag{2.12}$$

Lemma 2 *If the probability measure P on* $\mathcal{F}$ *is log-concave then P is also quasi-concave.*

Proof. Let $S_1, S_2 \in \mathcal{F}$ be convex sets such that $P(S_i) > 0$, $i = 1, 2$.[1] Now, since we are given that P is a log-concave probability measure, we have that for any $\lambda \in [0,1]$,

$$P(\lambda S_1 + (1-\lambda)S_2) \geq P^{\lambda}(S_1)P^{1-\lambda}(S_2)$$

Since the logarithm function is monotonic, if follows that

$$\begin{aligned} \ln P(\lambda S_1 + (1-\lambda)S_2) &\geq \lambda \ln P(S_1) + (1-\lambda)\ln P(S_2) \\ &\geq \min[\ln P(S_1), \ln P(S_2)] \end{aligned}$$

since any point on the line segment between two points must be at least as high as the lower of the two points. Hence

$$P(\lambda S_1 + (1-\lambda)S_2) \geq \exp(\min[\ln P(S_1), \ln P(S_2)]) = \min[P(S_1), P(S_2)]$$

and therefore P is a quasi-concave probability measure. ■

The following proposition answers the question of when a probability measure is log-concave or, more generally, quasi-concave. It is stated without proof because an advanced knowledge of measure theory is required - see Prékopa [38] and Borell [11].

Proposition 4 *Let the probability measure P on* $\Xi = \mathbb{R}^k$ *be of the continuous type with probability density function f. Then P is a log-concave measure iff f is a log-concave function (i.e.* $\ln f$ *is a concave function); and P is a quasi-concave measure iff* $f^{-1/k}$ *is a convex function.*

[1] Otherwise, i.e. if $P(S_1) = 0$ or $P(S_2) = 0$, the proof is trivial since $\min[P(S_1), P(S_2)] = 0 \leq P(\lambda S_1 + (1-\lambda)S_2) \because P(S) \geq 0 \, \forall S \in \mathcal{F}$.

2.3.3 Distributions that Can Lead to Convexity

We can now state the statistical distributions that have the desirable property of leading to a convex feasible set under appropriate assumptions on g. They are distributions with *quasi-concave probability measures*, such as those given in the following examples.

Example 1 *Uniform Distribution. For the k-dimensional uniform distribution on a convex body $S \subset \mathbb{R}^k$ with natural measure $\mu(S) > 0$ (see Appendix B.2.1), we have*

$$f^{-1/k}(\xi) = \left\{ \begin{array}{cc} \sqrt[k]{\mu(S)} & \textit{if } \xi \in S \\ \infty & \textit{if } \xi \notin S \end{array} \right]$$

which is a convex function in ξ and therefore the uniform distribution has a quasi-concave probability measure by Proposition 4.

Example 2 *Exponential Distribution. For the exponential distribution (see Appendix B.2.2), the logarithm of the density*

$$\ln f(\xi) = \left\{ \begin{array}{cc} -\infty & \textit{if } \xi < 0 \\ \ln \lambda - \lambda\xi & \textit{if } \xi \geq 0 \end{array} \right]$$

is a concave function in ξ, implying that the probability measure of the exponential distribution is log-concave (by Proposition 4) and hence quasi-concave (by Lemma 2).

Example 3 *Multivariate Normal Distribution. For the multivariate normal distribution (see Appendix B.2.3), the logarithm of the density*

$$\ln f(\xi) = -\frac{n}{2}\ln(2\pi) - \frac{1}{2}\ln|\Sigma| - \frac{1}{2}(\xi - \mu)^T\Sigma^{-1}(\xi - \mu)$$

is a concave function in ξ since Σ and hence its inverse Σ^{-1} are positive definite (refer to Sections 5.3 and 6.5 in Marlow [35]). Therefore the probability measure of the multivariate normal distribution is log-concave and hence quasi-concave.

It is mentioned in Section 3.2 of Birge & Louveaux [10] that the *multivariate t* distribution and the *multivariate F* distribution also have quasi-concave probability measures.

2.3.4 Closed Feasible Set

The following proposition, which is proved in Section 1.5 of Kall & Wallace [28], proves the closedness of the feasible set provided that $g(\cdot,\cdot)$ is jointly continuous in (x, ξ). This is important since for mathematical programs in general, we cannot assert the existence of solutions if the feasible sets are not known to be closed. The extreme value theorem (see Section 7.1 in Marlo [35]) states that when the feasible region is *closed* and bounded (*i.e.* compact), and the objective function is continuous, then a solution exists. When the feasible region is closed, it contains all limits of infinite sequences of points in the region, which is important when establishing global convergence of algorithms - see Chapter 6 in Luenberger [34].

Proposition 5 *If* $g : \mathbb{R}^n \times \Xi \to \mathbb{R}^m$ *is continuous then the feasible set* $B(\alpha)$ *is closed.*

2.4 Linear Chance-Constrained Problems

2.4.1 Linear Formulation

The chance-constrained problems (2.2) and (2.4) are classified as linear chance-constrained problems if the functions $g_i(x, \xi)$ are linear in x, and the set X is convex polyhedral. If we define $g_0(x, \xi) := c(\xi)^T x$, where $c(\xi) = (c_1(\xi), \ldots, c_n(\xi))^T$, then the objective function becomes

$$E_{\tilde{\xi}}\left[c(\tilde{\xi})^T x\right] = E_{\tilde{\xi}}\left[c(\tilde{\xi})\right]^T x = \bar{c}^T x \tag{2.13}$$

where $\bar{c} = E_{\tilde{\xi}}\left[c(\tilde{\xi})\right]$. This shows that when we are optimizing the expected value of a linear objective with uncertain coefficients, the *objective coefficients* can always be replaced by their expected values. If we define $g_i(x, \xi) := h_i(\xi) - T_i(\xi)x$ for $i = 1, \ldots, m$, where $h_i(\xi)$ is 1×1, $T_i(\xi)$ is $1 \times n$, and $X := \{x \in \mathbb{R}^n \mid Ax = b\,,\, x \geq 0\}$, then problem (2.2) becomes the ***stochastic linear program with joint chance constraints***

$$\left.\begin{aligned} &\min\ c^T x \\ &\text{s.t. } Ax = b \\ &\Pr\left[T(\tilde{\xi})x \geq h(\tilde{\xi})\right] \geq \alpha \\ &\qquad x \geq 0 \end{aligned}\right\} \tag{2.14}$$

where $T_i(\cdot)$ and $h_i(\cdot)$ denote the ith row of $T(\cdot)$ and the ith component of $\mathbf{h}(\cdot)$ respectively, so that $T(\cdot)$ is $m \times n$ and $\mathbf{h}(\cdot)$ is $m \times 1$. Problem (2.4) becomes the *stochastic linear program with separate chance constraints*

$$\left.\begin{array}{l} \min \mathbf{c}^T\mathbf{x} \\ \text{s.t. } A\mathbf{x} = \mathbf{b} \\ \Pr\left[T_i(\tilde{\xi})\mathbf{x} \geq h_i(\tilde{\xi})\right] \geq \alpha_i\,,\ i = 1,\ldots,m \\ \qquad \mathbf{x} \geq 0 \end{array}\right\} \tag{2.15}$$

2.4.2 Properties of Linear Chance-Constrained Programs

Returning to Proposition 3, the assumption of joint convexity of $\mathbf{g}(\cdot,\cdot)$ is so strong that it is not even satisfied in the linear cases (2.14) and (2.15) in general (since no assumption is made on the nature of the dependence on ξ), and hence Proposition 3 cannot be used to prove convexity of their feasible sets. In fact, problems (2.14) and (2.15) do not in general define convex feasible sets, although convexity can be asserted in special cases, as is shown in the following two propositions and in Sections 2.4.3 and 2.4.4.

Proposition 6 *The feasible set*

$$B(1) = \{\mathbf{x} \mid P(\{\xi \mid T(\xi)\mathbf{x} \geq \mathbf{h}(\xi)\}) = 1\} = \{\mathbf{x} \mid T(\xi)\mathbf{x} \geq \mathbf{h}(\xi)\ a.s.\}$$

is convex.

Proof. Assume that $\mathbf{x}, \mathbf{y} \in B(1)$ and $\lambda \in (0,1)$, so that $T(\xi)\mathbf{x} \geq \mathbf{h}(\xi)$ a.s. and $T(\xi)\mathbf{y} \geq \mathbf{h}(\xi)$ a.s. Let $\mathbf{z} = \lambda\mathbf{x} + (1-\lambda)\mathbf{y}$. Then $T(\xi)\mathbf{z} = \lambda T(\xi)\mathbf{x} + (1-\lambda)T(\xi)\mathbf{y} \geq \lambda\mathbf{h}(\xi) + (1-\lambda)\mathbf{h}(\xi)$ a.s. Thus $T(\xi)\mathbf{z} \geq \mathbf{h}(\xi)$ a.s. and therefore $\mathbf{z} \in B(1)$. ∎

Proposition 7 *Let $\tilde{\xi}$ have a finite discrete distribution described by* $\Pr\left[\tilde{\xi} = \xi^j\right] = p_j\,,\ j = 1,\ldots,r$*, where* $\sum_{j=1}^{r} p_j = 1$ *and* $p_j > 0\ \forall j$. *Then for* $\alpha > 1 - \min_j\{p_j\}$, *the feasible set*

$$B(\alpha) = \{\mathbf{x} \mid P(\{\xi \mid T(\xi)\mathbf{x} \geq \mathbf{h}(\xi)\}) \geq \alpha\}$$

is convex.

Proof. The support of the distribution is the finite set $\Xi = \{\xi^1, \ldots, \xi^r\}$. Therefore we ca write $\mathcal{B}(1) = \{\mathrm{x} \mid T(\xi)\mathrm{x} \geq \mathrm{h}(\xi)\ \forall \xi \in \Xi\}$. Consider an event A satisfying $P(A) = 1 - \min\{p_j\}$. If $p_k = \min\{p_j\}$ then $A = \bigcup_{j \neq k} \xi^j = \Xi - \{\xi^k\}$. Hence any event A' satisfying $P(A') > 1 - \min\{p_j\}$ must be $A' = A \cup \{\xi^k\} = \Xi$, since ξ^k is the only remaining realization with positive probabili y. Therefore for $\alpha > 1 - \min\{p_j\}$, $\mathcal{B}(\alpha) = \{\mathrm{x} \mid T(\xi)\mathrm{x} \geq \mathrm{h}(\xi)\ \forall \xi \in \Xi\} = \mathcal{B}(1)$ and hence $\mathcal{B}(\alpha)$ is convex by Proposition 6. ■

The above propositions show that stochastic linear programs with joint chance constra ts and discrete distributions, where the reliability levels are chosen sufficiently high, have c ex feasible sets. This is ... pply to stochastic linear programs with separate ch nce constraints where the ... h constraint is chosen sufficiently high.

2.4.3 Randomness ... de Only

We now consider the case wh ... x is fixed and there is randomness in the right-hand side only, *i.e.* $T(\xi) \equiv T$ and $\mathrm{h}(\xi) \equiv$ with the ith row of $T(\xi)$ and the ith e ment and $\mathrm{h}(\xi)$ being $T_i(\xi) \equiv T_i$ and $h_i(\xi) \equiv \xi_i$ respectively. For programs of the form (2.1), the joint chance constraints can be written as

$$\Pr\left[T\mathrm{x} \geq \tilde{\xi}\right] = F_{\tilde{\xi}}(T\mathrm{x}) \geq \alpha \tag{2.16}$$

where $F_{\tilde{\xi}}(\cdot)$ is the c.d.f. of $\tilde{\xi}$. Loosely speaking, the feasibility of x is equivalent to $T\mathrm{x}$ l ing in an upper $(1 - \alpha)$ confidence region for $\tilde{\xi}$.

Proposition 8 *Suppose that T is fixed and $\tilde{\xi}$ has an associated quasi-concave probabilit mea- sure. Then the feasible set of (2.14) is closed and convex.*

Proof. In this case, $\mathrm{g}(\mathrm{x}, \xi) = \xi - T\mathrm{x}$ which is continuous and linear in both x and ξ and hence jointly convex in (x, ξ). Therefore the feasible set is conve by Proposition 3 and closed by Proposition 5. ■

Proposition 9 *Suppose that T is fixed and the components $\tilde{\xi}_i$ of $\tilde{\xi}$ are independent random variables with distribution functions F_i. Then the feasible set can be written as*

$$B(\alpha) = \left\{ \mathbf{x} \mid \sum_i \ln F_i(T_i\mathbf{x}) \geq \ln \alpha \right\} \tag{2.17}$$

Furthermore, if the density functions f_i are log-concave, the feasible set is convex and closed.

Proof. By the independence of the $\tilde{\xi}_i$, $\Pr\left[T\mathbf{x} \geq \tilde{\xi}\right] = \prod_i \Pr\left[T_i\mathbf{x} \geq \tilde{\xi}_i\right] = \prod_i F_i(T_i\mathbf{x})$, so that $\Pr\left[T\mathbf{x} \geq \tilde{\xi}\right] \geq \alpha \Rightarrow \prod_i F_i(T_i\mathbf{x}) \geq \alpha \Rightarrow \sum_i \ln F_i(T_i\mathbf{x}) \geq \ln \alpha$, implying that $B(\alpha) = \{\mathbf{x} \mid \sum_i \ln F_i(T_i\mathbf{x}) \geq \ln \alpha\}$. If $\ln f_i$ is concave $\forall i$, it follows that $\sum_i \ln f_i$ is a concave function so that $\prod_i f_i$ is a log-concave function. Therefore the probability measure associated with $\tilde{\xi}$ is log-concave by Proposition 4, since its density function (which is $\prod_i f_i$ by independence of the $\tilde{\xi}_i$) is log-concave. Hence the feasible set is convex and closed by Proposition 8, since the probability measure is quasi-concave by Lemma 2. ■

For problems of the form (2.15), each separate chance constraint can be written in the form

$$\Pr\left[T_i\mathbf{x} \geq \tilde{\xi}_i\right] = F_i(T_i\mathbf{x}) \geq \alpha_i \Rightarrow T_i\mathbf{x} \geq F_i^{-1}(\alpha_i) \tag{2.18}$$

where F_i is the c.d.f. of $\tilde{\xi}_i$. Each chance constraint thus reduces to a linear constraint, and the deterministic equivalent of (2.15) is simply a linear program.

2.4.4 Normal Model for Separate Chance Constraints

In this section we show that for separate linear chance constraints, when the parameters in each constraint have a multivariate Normal distribution and are independent across constraints, the constraints lead to a convenient form and define a convex feasible set, so that the problem can be solved by convex nonlinear programming methods. Consider (2.15) with separate chance constraints of the form

$$\Pr\left[\tilde{\mathbf{T}}_i\mathbf{x} \geq \tilde{h}_i\right] \geq \alpha_i, \text{ where } \begin{pmatrix} \tilde{\mathbf{T}}_i & \tilde{h}_i \end{pmatrix}^T \sim N_{n+1}\left(\begin{bmatrix} \mu_{T_i} \\ \mu_{h_i} \end{bmatrix}, \begin{bmatrix} V_{T_i} & V_{T_i h_i} \\ V_{T_i h_i}^T & \sigma_{h_i}^2 \end{bmatrix} \right) \tag{2.19}$$

and where $\widetilde{\mathbf{T}}_i = T_i(\widetilde{\xi})$, $\widetilde{h}_i = h_i(\widetilde{\xi})$, $\mu_{T_i} = E\left(\widetilde{\mathbf{T}}_i^T\right)$, $V_{T_i} = \mathrm{Cov}\left(\widetilde{\mathbf{T}}_i^T\right)$, $\mu_{h_i} = E\left(\widetilde{h}_i\right)$, $\sigma_{h_i}^2 = \mathrm{Var}\left(\widetilde{h}_i\right)$ and $V_{T_ih_i} = \mathrm{Cov}\left(\widetilde{\mathbf{T}}_i^T, \widetilde{h}_i\right)$. We define the random variable

$$\widetilde{\zeta}_i := \widetilde{\mathbf{T}}_i\mathbf{x} - \widetilde{h}_i = \begin{pmatrix} \mathbf{x}^T & -1 \end{pmatrix} \begin{pmatrix} \widetilde{\mathbf{T}}_i & \widetilde{h}_i \end{pmatrix}^T \tag{2.20}$$

Since $\widetilde{\zeta}_i$ is a linear transformation of a random vector with a mul ariate Normal distribution, $\widetilde{\zeta}_i$ itself has a Normal distribution, with mean

$$E\left(\widetilde{\zeta}_i\right) = \begin{pmatrix} \mathbf{x}^T & -1 \end{pmatrix} \begin{pmatrix} \mu_{T_i} \\ \mu_{h_i} \end{pmatrix} = \mathbf{x}^T \mu_{T_i} - \mu_{h_i} =: m_i(\mathbf{x}) \tag{2.21}$$

and variance

$$\mathrm{Var}\left(\widetilde{\zeta}_i\right) = \begin{pmatrix} \mathbf{x}^T & -1 \end{pmatrix} \begin{pmatrix} V_{T_i} & V_{T_ih_i} \\ V_{T_ih_i}^T & \sigma_{h_i}^2 \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ -1 \end{pmatrix} = \mathbf{x}^T V_{T_i}\mathbf{x} - 2\mathbf{x}^T V_{T_ih_i} + \sigma_{h_i}^2 =: \sigma_i^2(\mathbf{x}) \tag{2.22}$$

The chance constraint (2.19) can then be reformulated as

$$\begin{aligned}
\Pr\left[\widetilde{\mathbf{T}}_i\mathbf{x} \geq \widetilde{h}_i\right] \;\geq\; & \alpha_i \Rightarrow \Pr\left[\widetilde{\zeta}_i \geq 0\right] \geq \alpha_i \\
\Rightarrow\; & \Pr\left[\frac{\zeta_i - m_i(\mathbf{x})}{\sigma_i(\mathbf{x})} \geq \frac{-m_i(\mathbf{x})}{\sigma_i(\mathbf{x})}\right] \geq \alpha_i \Rightarrow 1 - \Phi\left(\frac{-m_i(\mathbf{x})}{\sigma_i(\mathbf{x})}\right) \geq \alpha_i \\
\Rightarrow\; & \Phi\left(\frac{m_i(\mathbf{x})}{\sigma_i(\mathbf{x})}\right) \geq \alpha_i \Rightarrow \frac{m_i(\mathbf{x})}{\sigma_i(\mathbf{x})} \geq \Phi^{-1}(\alpha_i)\,, \text{ since } 1 - \Phi(-z) \equiv \Phi(z) \\
\Rightarrow\; & \sigma_i(\mathbf{x})\Phi^{-1}(\alpha_i) - m_i(\mathbf{x}) \leq 0
\end{aligned} \tag{2.23}$$

Proposition 10 *The standard deviation $\sigma_i(\mathbf{x})$ is a convex function of $\mathbf{x}$.*

Proof. We prove that $\sigma_i(\mathbf{x})$ is convex in $\mathbf{x}$ by proving that $g(\mathbf{x}) = \sqrt{\mathbf{x}^T \Sigma \mathbf{x}}$ is co. . in $\mathbf{x}$, where $\Sigma > O$. Since Σ is real symmetric and positive definite, it can be diagonalized as

$$\Sigma = U^T \Delta U = U^T \Delta^{1/2} \Delta^{1/2} U = \left(\Delta^{1/2}U\right)^T \left(\Delta^{1/2}U\right)$$

where $\Delta = \text{diag}(\delta_1, \ldots, \delta_n)$, $\delta_i > 0 \; \forall i$ and $\Delta^{1/2} = \text{diag}\left(\sqrt{\delta_1}, \ldots, \sqrt{\delta_n}\right)$. Now by the Cauchy-Schwartz inequality,

$$\begin{aligned}\sqrt{\mathbf{x}_1^T \Sigma \mathbf{x}_1}\sqrt{\mathbf{x}_2^T \Sigma \mathbf{x}_2} &= \sqrt{\mathbf{x}_1^T \left(\Delta^{1/2}U\right)^T \left(\Delta^{1/2}U\right) \mathbf{x}_1}\sqrt{\mathbf{x}_2^T \left(\Delta^{1/2}U\right)^T \left(\Delta^{1/2}U\right) \mathbf{x}_2} \\ &= \left\|\Delta^{1/2}U\mathbf{x}_1\right\| \left\|\Delta^{1/2}U\mathbf{x}_2\right\| \geq \left(\Delta^{1/2}U\mathbf{x}_1\right)^T \left(\Delta^{1/2}U\mathbf{x}_2\right) = \mathbf{x}_1^T \Sigma \mathbf{x}_2\end{aligned}$$

Therefore for any $\lambda \in (0,1)$,

$$2\lambda(1-\lambda)\sqrt{\mathbf{x}_1^T \Sigma \mathbf{x}_1}\sqrt{\mathbf{x}_2^T \Sigma \mathbf{x}_2} \geq 2\lambda(1-\lambda)\mathbf{x}_1^T \Sigma \mathbf{x}_2$$

Adding $\lambda^2 \mathbf{x}_1^T \Sigma \mathbf{x}_1 + (1-\lambda)^2 \mathbf{x}_2^T \Sigma \mathbf{x}_2$ to each side gives

$$\begin{aligned}\text{LHS} &= \lambda^2 \mathbf{x}_1^T \Sigma \mathbf{x}_1 + (1-\lambda)^2 \mathbf{x}_2^T \Sigma \mathbf{x}_2 + 2\lambda(1-\lambda)\sqrt{\mathbf{x}_1^T \Sigma \mathbf{x}_1}\sqrt{\mathbf{x}_2^T \Sigma \mathbf{x}_2} \\ &= \lambda^2 [g(\mathbf{x}_1)]^2 + (1-\lambda)^2 [g(\mathbf{x}_2)]^2 + 2\lambda(1-\lambda)g(\mathbf{x}_1)g(\mathbf{x}_2) = [\lambda g(\mathbf{x}_1) + (1-\lambda)g(\mathbf{x}_2)]^2\end{aligned}$$

and

$$\begin{aligned}\text{RHS} &= \lambda^2 \mathbf{x}_1^T \Sigma \mathbf{x}_1 + (1-\lambda)^2 \mathbf{x}_2^T \Sigma \mathbf{x}_2 + \lambda(1-\lambda)\mathbf{x}_1^T \Sigma \mathbf{x}_2 + \lambda(1-\lambda)\mathbf{x}_2^T \Sigma \mathbf{x}_1 \\ &= (\lambda\mathbf{x}_1 + (1-\lambda)\mathbf{x}_2)^T \Sigma (\lambda\mathbf{x}_1 + (1-\lambda)\mathbf{x}_2) = [g(\lambda\mathbf{x}_1 + (1-\lambda)\mathbf{x}_2)]^2\end{aligned}$$

Therefore $g(\lambda\mathbf{x}_1 + (1-\lambda)\mathbf{x}_2) \leq \lambda g(\mathbf{x}_1) + (1-\lambda)g(\mathbf{x}_2)$ since $g(\cdot)$ is always non-negative, and hence g is a convex function of $\mathbf{x}$. Since the vector $\left(\widetilde{\mathbf{T}}_i, \widetilde{h}_i\right)^T$ is normally distributed, its variance-covariance matrix is required to be positive definite (see Appendix B.2.3), and hence it follows that $\sigma_i(\mathbf{x})$ is a convex function of $\mathbf{x}$. ∎

The constraint (2.19) defines a convex feasible set if the LHS of (2.23) is convex in $\mathbf{x}$. Since $m_i(\mathbf{x})$ is affine in $\mathbf{x}$ and $\sigma_i(\mathbf{x})$ is convex in $\mathbf{x}$ by the above proposition, the LHS is convex iff $\Phi^{-1}(\alpha_i) \geq 0$ which is true iff $\alpha_i \geq 0.5$. Thus we have, under the assumption of normality and $\alpha_i \geq 0.5 \; \forall i$, a deterministi convex program that can be solved by standard methods of nonlinear programming.

Table 2.1: Mean and Variance Functions

i	m_i	σ_i^2	α_i
1	$2x_1 + 3x_2 + 5x_3 - 10$	$9x_1^2 + 4x_2^2 + 16x_3^2 + 12x_1x_3$	0.95
2	$7x_1 + 5x_2 + x_3 - 60$	25	0.90

Example

Consider the stochastic linear program with two separate chance constraints

$$\left.\begin{array}{ll} \min & 5x_1 + 3x_2 + x_3 \\ \text{s.t.} & \Pr\left[\tilde{\xi}_1 x_1 + \tilde{\xi}_2 x_2 + \tilde{\xi}_3 x_3 \geq 10\right] \geq 0.95 \\ & \Pr\left[7x_1 + 5x_2 + x_3 \geq \tilde{\xi}_4\right] \geq 0.9 \\ & x_1, x_2, x_3 \geq 0 \end{array}\right\} \tag{2.24}$$

where $\begin{pmatrix} \tilde{\xi}_1 \\ \tilde{\xi}_2 \\ \tilde{\xi}_3 \end{pmatrix} \sim N_3\left(\begin{bmatrix} 2 \\ 3 \\ 5 \end{bmatrix}, \begin{bmatrix} 9 & 0 & 6 \\ 0 & 4 & 0 \\ 6 & 0 & 16 \end{bmatrix}\right)$ and $\tilde{\xi}_4 \sim N(60, 25)$ independently. The problem has mean and variance functions as given in Table 2.1, so that its deterministic equivalent is the convex program

$$\left.\begin{array}{ll} \min & z = 5x_1 + 3x_2 + x_3 \\ \text{s.t.} & \sqrt{9x_1^2 + 4x_2^2 + 16x_3^2 + 12x_1x_3}\,\Phi^{-1}(0.95) - (2x_1 + 3x_2 + 5x_3 - 10) \leq 0 \\ & 5\Phi^{-1}(0.90) - (7x_1 + 5x_2 + x_3 - 60) \leq 0 \\ & x_1, x_2, x_3 \geq 0 \end{array}\right\} \tag{2.25}$$

with the optimal solution obtained by MATLAB (see Appendix A.5) as

$$\left.\begin{array}{l} z^* = 42.004805 \\ x_1^* = 0 \\ x_2^* = 12.2014764 \\ x_3^* = 5.4003759 \end{array}\right\} \tag{2.26}$$

Table 2.2: Comparison of Solutions

	p_1	p_2	z
Stochastic solution	0.95	0.90	42.0048051
Deterministic solution	0.8606698	0.5	36

If we had ignored the variability in the problem, we would have solved the deterministic LP

$$\left.\begin{aligned} \min\ & z = 5x_1 + 3x_2 + x_3 \\ \text{s.t.}\ & 2x_1 + 3x_2 + 5x_3 \geq 10 \\ & 7x_1 + 5x_2 + x_3 \geq 60 \\ & x_1, x_2, x_3 \geq 0 \end{aligned}\right\} \tag{2.27}$$

which leads to the deterministic solution

$$\left.\begin{aligned} z^* &= 36 \\ x_1^* &= 0 \\ x_2^* &= 12 \\ x_3^* &= 0 \end{aligned}\right\} \tag{2.28}$$

At first sight, the deterministic solution appears to be better since it leads to a lower value of the objective. However, when we calculate $p_i = \Pr[i\text{th constraint is satisfied}]$ for $i = 1, 2$, as shown in Table 2.2, we see that the deterministic solution leads to much lower reliability levels. In fact, the probability that the deterministic solution satisfies both constraints is only $0.8606698 \times 0.5 \simeq 43\%$, while the reliability of the stochastic solution is $0.95 \times 0.90 = 85.5\%$. The stochastic solution improves reliability at the expense of the objective. Note that both of the constraints are active at the stochastic solution. This example clearly indicates that when reliability is an important criterion (in addition to the cost in the objective), the stochastic solution must be used.

2.5 Solution Methods

As has been shown in Sections 2.3 and 2.4, chance-constrained problems such as (2.2), (2.4), (2.14) or (2.15) may define smooth convex mathematical programing problems, at least under appropriate assumptions. Sometimes these programs have a special form and thus standard methods of nonlinear or even linear programming can be used. In general, however, if we wanted to solve these problems by standard nonlinear programming methods, we would have to obtain gradients and evaluations for functions involving expectations and probabilities, such as $E_{\tilde{\xi}}\left[g_0\left(x,\tilde{\xi}\right)\right]$ and $P\left(\{\xi \mid g(x,\xi) \leq 0\}\right) \geq \alpha$. Each evaluation of these functions requires multivariate integration, which usually cannot be computed exactly and may even be numerically intractable for problems of moderate dimension, *i.e.* problems with several random variables. In practice, however, it is fairly common for problems of high dimension (*i.e.* with hundreds or thousands of random variables) to arise, and therefore many chance constrained problems are not efficiently solvable, except under simplifying assumptions. Various computational methods have been devised that try to avoid the exact evaluation of the multivariate integration that appears in these problems. See, for example, the PROCON procedure in Section 4.1 of Kall & Wallace [28] that solves programs of the form (2.14) with T fixed and h normally distributed.

A further problem in solving chance-constrained programs is that chance constraints can easily define a nonconvex feasible set. This leads to severe computational problems in finding a global optimum. Proposition 7 also shows that in solving chance-constrained problems, we cannot expect the approach of approximating a continuous distribution by successively refined discrete distributions to be successful. This is because successive refinements of the discrete distribution would eventually imply that $\min\{p_j\} < 1-\alpha$, so that the approximating problems could become nonconvex, even if the original problem (with its continuous distribution) were convex. And, of course, solutions methods should not entail replacing convex problems by nonconvex ones.

2.6 Example: Water Resource Management

This section provides an illustrative example in water resource management that was adapted from an exercise at the end of Chapter 1 of Birge & Louveaux [10]. The example illustrates various alternative stochastic models that can be formulated, such as a P-model, a recourse model and a chance-constrained model. It is also shown that for this problem, deterministic models do not aid decision making. The solutions and relative merits of the various models are compared under different distributional assumptions.

2.6.1 The Water Level Problem

Consider the water level in a dam, which can be controlled by *releasing water* through the sluices. In this manner, it is possible to lower the level of the dam by at most 200 mm per month. However, the level of the dam is affected by other important factors that increase the water level, such as *rainfall*, and factors that lower the water level, such as *evaporation* and *extraction* (for consumption). Suppose that evaporation and extraction together account for a lowering of the water level at a constant rate of 50 mm per month. This is a simplifying assumption, since evaporation depends on temperature, humidity and cloud cover which are all random, and the amount extracted depends on the consumer demand for water. We assume, however, that the rainfall over the next month is not known with certainty, as the weather in the catchment area is variable.

Suppose that the water level in the dam is currently 150 mm below the flood level and 100 mm above the shortage level. If *flooding* occurs, damage will be caused to properties and towns below the dam wall. This damage is assessed at R10,000 per mm above the flood level. On the contrary, a *shortage* leads to the costly importation of water. The cost of this importation is estimated at R5,000 per mm below the shortage level. We assume that the cost of releasing water through the sluices is negligible.

2.6.2 Modelling the Problem

We are trying to decide how much water to release over the next month. A balance is sought between releasing too much water which can result in a shortage, and releasing too little water which can result in flooding. Our goal could be to minimize the expected costs arising from flooding or shortage, or to minimize the probability of a flood or shortage. Our *decision variable* is x and the *uncertainty* in the problem is characterized by the random variable $\tilde{\xi}$, where we define

$x =$ the amount (in mm) by which the water level should be lowered during the next month, restricted to $0 \leq x \leq 200$

$\tilde{\xi} =$ the amount (in mm) of rainfall during the next month, $0 \leq \tilde{\xi} < \infty$

$F(\cdot) =$ the c.d.f. of $\tilde{\xi}$

It is estimated that the distribution of rainfall over the next month has mean $E(\widetilde{\xi}) = 188$ and variance $\text{Var}(\widetilde{\xi}) = 12996$. We compare the results obtained using four different distributions for the rainfall, each of which has the same mean and variance (see Appendix B.2).

1. The Gamma distribution, $\widetilde{\xi} \sim G\,(r = 2.719606,\ \lambda = 0.01446599)$
2. The Lognormal distribution, $\widetilde{\xi} \sim LN\left(\mu = 5.079877,\ \sigma^2 = 0.3131307\right)$
3. The Weibull distribution, $\widetilde{\xi} \sim W\,(\alpha = 1.697216,\ \beta = 210.6809)$
4. A mixture of the Point distribution at $\xi = 0$, with probability 1/8 and the Weibull distribution on $\xi \in (0, \infty)$, with probability 7/8. Therefore $\widetilde{\xi} \sim \frac{1}{8}Pt(0) + \frac{7}{8}W\,(2.401723, 242.3673)$. This distribution has the mass and density function

$$f(\xi) = \left\{ \begin{array}{cll} \frac{1}{8} & \text{if } \xi = 0 & \text{(mass)} \\ \frac{7}{8}ab^{-a}\xi^{a-1}e^{-(\xi/b)^a} & \text{if } \xi > 0 & \text{(density)} \end{array} \right] \tag{2.29}$$

and the distribution function

$$F(\xi) = \left\{ \begin{array}{cl} 0 & \text{if } \xi < 0 \\ 1 - \frac{7}{8}\exp\left\{-\left(\frac{\xi}{b}\right)^a\right\} & \text{if } \xi \geq 0 \end{array} \right] \tag{2.30}$$

The first three distributions are standard continuous distributions with non-negative support. They all have continuous distribution functions and continuous density functions, with $f(0) = F(0) = 0$. The fourth distribution has been used as a realistic distribution to model rainfall in [16]. This distribution assumes that there is a positive probability of having zero rainfall in the month and that the amount of rainfall has a Weibull distribution given that there is some rainfall in the month. Its distribution and density functions both have a discontinuity at $\xi = 0$.

2.6.3 *P*-model

The water level at the end of the month will have increased by $\xi - x - 50$, where ξ is the realization of $\widetilde{\xi}$. A flood will occur if this increase is greater than 150 mm and a shortage will occur if the decrease is greater than or equal to 100 mm. We adopt the convention that flooding occurs if the flood level is exceeded, while shortage occurs if the shortage level is equalled or "exceeded". Therefore

$$\begin{aligned} \Pr[\text{Flooding}] &= \Pr\left[\tilde{\xi} - x - 50 > 150\right] = \Pr\left[\tilde{\xi} > x + 200\right] = 1 - F(x + 200) && (2.31) \\ \Pr[\text{Shortage}] &= \Pr\left[\tilde{\xi} - x - 50 \le -100\right] = \Pr\left[\tilde{\xi} \le x - 50\right] = F(x - 50) && (2.32) \end{aligned}$$

We could minimize the probability of flooding to obtain

$$\min_{0 \le x \le 200} 1 - F(x + 200) \Rightarrow \max_{0 \le x \le 200} (x + 200) \Rightarrow x = 200$$

or minimize the probability of shortage to get

$$\min_{0 \le x \le 200} F(x - 50) \Rightarrow 0 \le x < 50, \text{ since } F(u) = 0\ \forall u < 0$$

but unfortunately, minimizing the probability of flooding maximizes the probability of shortage and *vice versa*. We therefore create a model that minimizes the probability of flooding *or* shortage. Since $\Pr[\text{Flooding or Shortage}] = \Pr[\text{Flooding}] + \Pr[\text{Shortage}]$, our model becomes

$$\min_{0 \le x \le 200} P(x) = 1 - F(x + 200) + F(x - 50) \qquad (2.33)$$

We are minimizing a function of one variable on the closed interval $x \in [0, 200]$, and therefore the minimum is attained either at $x^* = 0$ or 200, or at a point x^* that satisfies

$$P'(x) = f(x - 50) - f(x + 200) = 0 \Rightarrow f(x^* - 50) = f(x^* + 200) \qquad (2.34)$$

where $f(\cdot)$ is the p.d.f. of $\tilde{\xi}$, provided that P is continuously differentiable on $[0, 200]$. However, this is not true for all of the distributions, since if $F(\xi)$ has a discontinuity at $\xi = 0$, $P(x)$ will have a discontinuity at $x = 50$, and if $f(\xi)$ has a discontinuity at $\xi = 0$, $P(x)$ will not be differentiable at $x = 50$. Since we only consider distributions with potential discontinuities at 0, the minimum will be attained at either $x^* \in \{0, 50, 200\}$ or x^* satisfying (2.34). The functions $P(x)$ for each distribution are illustrated in Figure 2-1 and the computational results for the P-model (which were obtained by MATLAB - see Appendix A.5) are given in Table 2.3. We

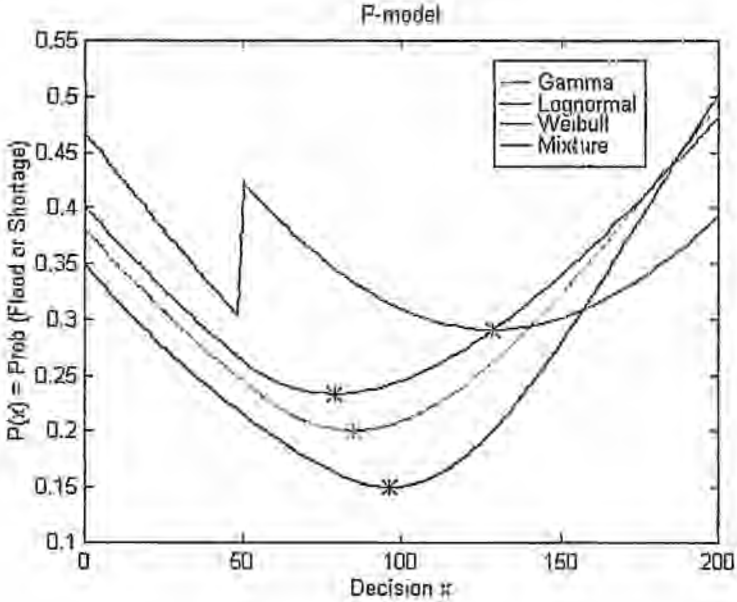

Figure 2-1: P-Model Objective for each Distribution

denote the optimal decision under the P-model as x_P^*.

2.6.4 E-Model

Instead of using the probability of flooding or shortage as our objective, we could minimize the expected cost of flooding or shortage. The amount of flooding is $\max(\xi - x - 200, 0)$ causing damage to the value of $10\max(\xi - x - 200, 0)$ and the amount of shortage is $\max(x - \xi - 50, 0)$ incurring an importation cost of $5\max(x - \xi - 50, 0)$, where we work in units of $R1,000$. If we

Table 2.3: Computational Results for the P-model

Distribution	Optimal x_P^*	Pr[Shortage]	Pr [Flood]	$P(x_P^*) = \min P(x)$	$\max P(x)$
Gamma	84.763	17.57%	2.52%	20.08%	49.30% at 200
Lognormal	96.580	13.69%	1.34%	15.03%	50.24% at 200
Weibull	79.077	19.96%	3.41%	23.37%	48.12% at 200
Mixture	128.404	10.99%	18.13%	29.12%	46.58% at 0

define $E(x)$ as the expected cost of flooding or shortage (which we want to minimize) associated with decision x, we obtain the program

$$\min_{0\leq x\leq 200} E(x) = 10E_{\tilde{\xi}}\left[\max(\tilde{\xi} - x - 200, 0)\right] + 5E_{\tilde{\xi}}\left[\max(x - \tilde{\xi} - 50, 0)\right] \tag{2.35}$$

Program (2.35) has the form of an E-model without chance constraints. In fact, it is a simple recourse model where the recourse variables are the amount of flooding and shortage respectively (see Chapter 3), and the penalty costs are the damage and importation costs. Note that to form such a model it was necessary to quantify these penalty costs. Now

$$\begin{aligned} E(x) &= 10\int_0^\infty \max(\xi - x - 200, 0) f(\xi) d\xi + 5\int_0^\infty \max(x - \xi - 50, 0) f(\xi) d\xi \\ &= \left\{ \begin{array}{ll} 10\int_{x+200}^\infty (\xi - x - 200) f(\xi) d\xi & \text{if } 0 \leq x < 50 \\ 10\int_{x+200}^\infty (\xi - x - 200) f(\xi) d\xi + 5\int_0^{x-50} (x - \xi - 50) f(\xi) d\xi & \text{if } 50 \leq x \leq 200 \end{array} \right] \end{aligned} \tag{2.36}$$

The function $E(x)$ is continuous on $[0, 200]$ for all four distributions. After using Leibnitz's rule for differentiation under the integral sign[2] and simplifying, we obtain

$$E'(x) = \left\{ \begin{array}{ll} 10F(x+200) - 10 & \text{if } 0 \leq x < 50 \\ 10F(x+200) - 10 + 5F(x-50) & \text{if } 50 < x \leq 200 \end{array} \right] \tag{2.37}$$

and differentiating again,

$$E''(x) = \left\{ \begin{array}{ll} 10f(x+200) & \text{if } 0 \leq x < 50 \\ 10f(x+200) + 5f(x-50) & \text{if } 50 < x \leq 200 \end{array} \right] \tag{2.38}$$

The expressions (2.37) and (2.38) also apply for $E'(50)$ provided that $F(0) = 0$ and for $E''(50)$ provided that $f(0) = 0$. This is true for the three continuous distributions but not for the mixture distribution. Therefore $E(x)$ has a non-differentiable point for the mixture distribution at $x = 50$, while for the other distributions $E(x)$ is twice continuously differentiable on $[0, 200]$. For all four distributions, $E(x)$ is convex since $E''(x) > 0\ \forall x \in [0, 200]$ and $E'(x)$ has a jump

[2] $\frac{d}{dx}\int_{\ell(x)}^{u(x)} g(x,\xi)d\xi = \int_{\ell(x)}^{u(x)} \frac{\partial g(x,\xi)}{\partial x} d\xi + g(x, u(x))\, u'(x) - g(x, \ell(x))\, \ell'(x)$

Table 2.4: Computational Results for the E-model

Distribution	Optimal x_E^*	$E(x_E^*) = \min E(x)$
Gamma	141.517	120.323
Lognormal	144.803	117.818
Weibull	139.977	118.750
Mixture	156.823	66.383

increase at $x = 50$ for the mixture distribution. Consider the interval $0 \le x < 50$. Since $E'(0) = 10F(200) - 10 < 0$ and $E'(50) = 10F(250) - 10 < 0$ and $E(x)$ is a convex function of x on $[0, 50)$, the minimum of $E(x)$ cannot be attained on this interval. Next, consider the interval $50 \le x \le 200$. The minimum of $E(x)$ on $[50, 200]$ will occur at $x^* = 50$ or $x^* = 200$ or at x^* satisfying $E'(x^*) = 0$. The functions $E(x)$ for each distribution are plotted on $x \in [50, 200]$ in

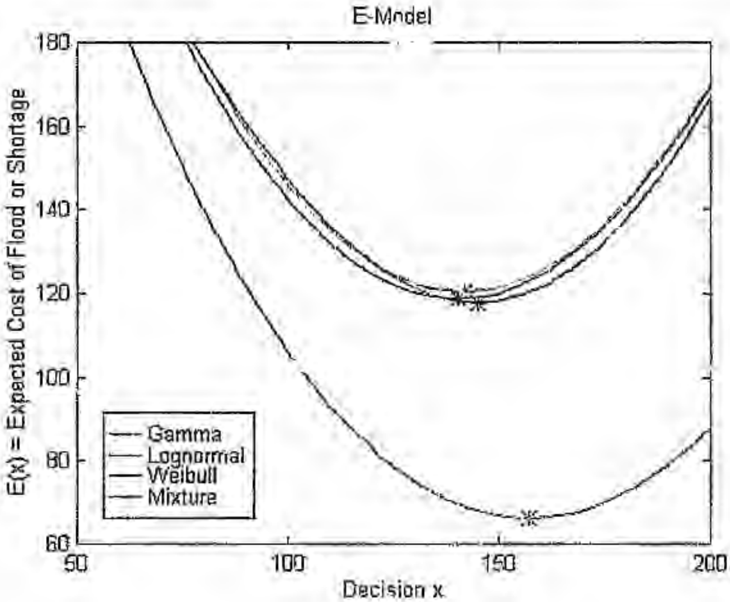

Figure 2-2: E-Model Objective for each Distribution

Figure 2-2 and the computational results for the E-model (which were obtained by MATLAB) are given in Table 2.4. We denote the optimal solution under the E-model as x_E^*.

2.6.5 Deterministic Models

The deterministic versions of the P-model and E-model can be obtained by putting $\widetilde{\xi} = 188$ with probability 1, *i.e.* $\widetilde{\xi} \sim Pt(188)$, with the c.d.f.

$$F(\xi) = \begin{Bmatrix} 0 & \text{if } \xi < 188 \\ 1 & \text{if } \xi \geq 188 \end{Bmatrix} \tag{2.39}$$

It follows that $P(x) = 1 - F(x+200) + F(x-50) = 1 - 1 + 0 = 0\ \forall x \in [0, 200]$. The P-model thus becomes degenerate in this case, as no value of x can lead to flooding or shortage. In the E-model, the amount of flooding is $\max(188 - x - 200, 0) = \max(-x - 12, 0) = 0\ \forall x \in [0, 200]$ and the amount of shortage is $\max(x - 188 - 50, 0) = \max(x - 238, 0) = 0\ \forall x \in [0, 200]$. Therefore $E(x) = 0\ \forall x \in [0, 200]$, so that the E-model is also degenerate in this case. In these deterministic models, any decision on x is optimal. Clearly, this is misleading and in this example, deterministic models have absolutely *no value* in helping us to determine a decision.

2.6.6 Chance-Constrained Model

So far we have considered a P-model that minimizes the probability of a disaster (*i.e.* flooding or shortage) occurring without regard to the cost of the disaster. We also considered an E-model in the form of a pure recourse model that minimizes the expected cost of disaster without regard to its probability of occurrence. In this section, we create a *chance-constrained model* that lies somewhere between these two models. It minimizes the expected cost of disaster, subject to the constraint that the probability of a disaster occurring must be below a certain level α. In the sense of probabilistic programming, this model is an E-model with a chance constraint.

$$\begin{aligned} \min E(x) &= E_{\widetilde{\xi}}\left[10\max(\widetilde{\xi} - x - 200, 0) + 5\max(x - \widetilde{\xi} - 50, 0)\right] && (2.40a) \\ \text{s.t. } P(x) &= 1 - F(x+200) + F(x-50) \leq \alpha && (2.40b) \\ 0 &\leq x \leq 200 \end{aligned}$$

Table 2.5: Computational Results for the Chance-Constrained Model

Distribution	x_ℓ	x_u	x_C^*	$E(x_C^*)$	$P(x_C^*)$
Gamma	27.392	142.270	141.517	120.323	29.78%
Lognormal	15.578	154.606	144.803	117.818	25.90%
Weibull	35.031	133.182	133.182	119.431	30%
Mixture	108.660	148.416	148.416	67.228	30%

Note that $E(x)$ is the same as (2.35) and $P(x)$ is given by (2.33). We choose $\alpha = 0.30$ and refer to Figure 2-1. For the three continuous distributions, $P(x)$ decreases from $P(0)$ to a minimum $P(x_P^*)$ and then increases to $P(200)$. For the mixture distribution, $P(x)$ decreases from $P(0)$ towards $P(50)$ but does not cross the level 0.30. There is a jump increase to $P(50)$ and the function then exhibits the same behaviour as for the other distributions. Therefore, for each distribution, the chance constraint $P(x) \leq 0.30$ leads to the constraint $x_\ell \leq x \leq x_u$, so that the chance-constrained model is equivalent to minimizing $E(x)$ over an interval. We denote the solution to the chance-constrained problem as x_C^*. The computational results for the chance-constrained model (which were obtained by MATLAB) are given in Table 2.5. Note that $x_C^* = x_E^*$ when the chance constraint is inactive, *i.e.* when $P(x_C^*) < 0.30$.

2.6.7 Sensitivity to Changes in α

In this section, we show how to calculate the value of x_C^* for different values of $\alpha \in [0, 1]$, and illustrate the results. For the three continuous distributions, the chance constraint $P(x) \leq \alpha$ implies $x_\ell(\alpha) \leq x \leq x_u(\alpha)$. We follow these steps to calculate $x_\ell(\alpha)$ and $x_u(\alpha)$:

1. If $\alpha < P(x_P^*)$ then x_ℓ and x_u do not exist and there is no feasible solution x_C^*

2. If $\alpha = P(x_P^*)$ then $x_\ell = x_u = x_P^*$

3. If $\alpha > P(x_P^*)$ then x_ℓ and x_u are calculated as follows:

   (a) if $\alpha \geq P(0)$ then $x_\ell = 0$, otherwise x_ℓ solves $P(x_\ell) = \alpha$, $0 < x_\ell < x_P^*$

   (b) if $\alpha \geq P(200)$ then $x_u = 200$, otherwise x_u solves $P(x_u) = \alpha$, $x_P^* < x_u < 200$

Once we have found $x_l(\alpha)$ and $x_u(\alpha)$, we calculate x_C^* as

$$x_C^* = \begin{cases} x_E^* & \text{if } x_E^* \in [x_l, x_u] \\ \arg\min\left(E(x_l), E(x_u)\right) & \text{otherwise} \end{cases} \tag{2.41}$$

Note that we utilized the optimal solutions x_P^* and x_E^* from the P-model and the E-model. This indicates that the three models are very closely related. For the mixture distribution, the procedure to calculate x_l and x_u changes slightly. In this case, the solution once again cannot occur in $[0, 50)$ and so we follow the same procedure as above, replacing the interval $[0, 200]$ by $[50, 200]$. Figure 2-3 plots the value of x_C^* against α for each distribution. The graph shows that the solution x_C^* remains constant (at x_E^*) once α reaches a certain level, which is $P(x_E^*)$. In other words, the chance constraint becomes inactive $\forall\alpha \geq P(x_E^*)$, leading to the constant decision $x_C^* = x_E^*$ for all values of α on this interval.

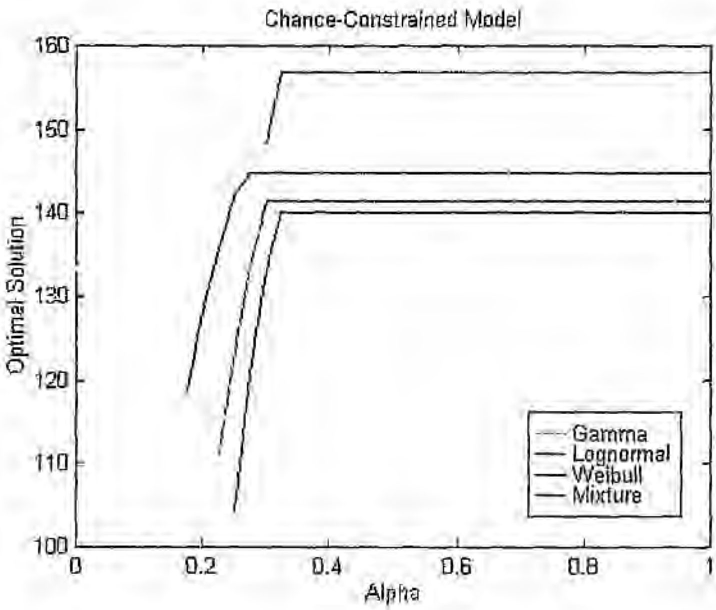

Figure 2-3: Sensitivity of the Solution to Changes in α

Table 2.6: Comparison of Solutions under Different Models

Distribution	$P(x_P^*)$	$P(x_E^*)$	$P(x_C^*)$	$E(x_P^*)$	$E(x_E^*)$	$E(x_C^*)$
Gamma	20.08%	29.78%	29.78%	167.837	120.323	120.323
Lognormal	15.03%	25.90%	25.90%	151.740	117.818	117.818
Weibull	23.37%	31.53%	30%	174.318	118.7 [illegible]	119.431
Mixture	29.12%	30.87%	30%	76.148	66.383	67.228

2.6.8 Comparison of Models

Which solution do we use? Table 2.6 shows the value of the optimal solution of each model for each distribution in terms of expected cost and probability of disaster. There is no solution that is optimal for all of the models. When we minimize $P(x)$ or $E(x)$, we are using different criteria and different optimal solutions result. The optimal decision x_P^* under the P-model is suboptimal under the E-model and the optimal decision x_E^* under the E-model is suboptimal under the P-model. The chance-constrained model seeks a balance between the two but is often suboptimal under both models.

The best decision really depends on the preferences of the decision maker. There is no mathematical reason to prefer one model to the next. The decision maker must decide which criteria are the most important.

Chapter 3

Stochastic Linear Programming with Recourse

This chapter contains a development of two-stage stochastic linear programs with recourse. Multistage problems are mentioned but not discussed in detail. The chapter begins with a discussion of the concept of recourse and recourse models. The formulation and properties of linear recourse programs are then described. Solution methods and issues are discussed and a worked example is provided in which solution methods and alternative models are compared. The chapter concludes with a comparison between recourse models and chance-constrained models in general.

3.1 Recourse Problems

Stochastic linear programs are linear programs in which some of the problem data or parameters are uncertain. *Recourse programs* are mathematical programs in which decisions or recourse actions can be taken after the uncertainty has been disclosed. In this context, data uncertainty means that the problem data can be expressed in terms of random variables. We assume that an accurate probabilistic description of these random variables is available in the form of distribution- or density functions. The particular values that the random variables take on can be seen as the set of possible results of a random experiment. With respect to this random experiment, the set of decisions to be taken can be classified into two groups:

- Decisions that must be taken *before* the random experiment. These decisions, denoted by $\mathbf{x}$, are called *first-stage decisions* and the period during which they are taken is called the *first stage*.

- Decisions that must be taken *after* the random experiment. These decisions, denoted by $\mathbf{y}$, are called *second-stage decisions* and the corresponding period is called the *second stage*.

If the result of the random experiment is $\omega \in \Omega$ where Ω is the sample space of the experiment, the sequence of decisions and events can be represented diagrammatically as

$$\mathbf{x} \to \xi(\omega) \to \mathbf{y}(\omega, \mathbf{x})$$

This shows that the second-stage decisions are functions of the outcome of the random experiment and of the first-stage decision. It is important to understand that the definitions of first stage and second stage refer only to whether the decisions precede or follow the random experiment, and thus stages can include periods of time where sequential decisions are taken.

Many practical decision problems involve a *sequence* of decisions which react to outcomes that evolve over time. These problems cannot simply be modelled as two-stage stochastic programs. If a sequence of random vectors can be observed at different times in the study horizon, and a sequence of decisions is taken in reaction to these observations, then the problem has the form of a *multistage stochastic program*. The transition between stages is made when a new observation of a random vector is made. Stages can therefore be periods of time that are very different in length. The concept of *nonanticipativity* in multistage programs requires that decisions made at a certain stage:

- can depend on decisions that have already been taken and on the outcomes of random vectors that have already been observed, but

- must be independent of decisions that are yet to be taken and independent of outcomes of random vectors that are yet to be observed.

The sequence of actions in a multistage problem with t stages is as follows:

- Take the first-stage decision $\mathbf{x}^1$
- Observe the random vector ξ^1
- Take the second-stage decision $\mathbf{x}^2 = \mathbf{x}^2(\xi^1 ; \mathbf{x}^1)$
- Observe the random vector ξ^2
- Take the third-stage decision $\mathbf{x}^3 = \mathbf{x}^3(\xi^1, \xi^2 ; \mathbf{x}^1, \mathbf{x}^2)$

$\vdots \quad \vdots \quad \vdots \quad \vdots$

- Observe the random vector ξ^{t-1}
- Take the tth-stage decision $\mathbf{x}^t = \mathbf{x}^t(\xi^1, \ldots, \xi^{t-1} ; \mathbf{x}^1, \ldots, \mathbf{x}^{t-1})$

In the two-stage recourse problem, we need to choose $\mathbf{x}$ before we know the realized value of $\tilde{\xi}$, and once we know this value, we cannot just change $\mathbf{x}$ accordingly. Ideally, we would like to choose the optimal value of $\mathbf{x}$ once we knew the realized value of $\tilde{\xi}$, so that the best possible decision could be made. Such a decision is known as a *wait-and-see* solution. Unfortunately, wait-and-see solutions are not what we need. We have to decide $\mathbf{x}$ under uncertainty and take what is known as a *here-and-now* decision.

3.2 Formulation of Two-Stage Linear Recourse Problems

It was discussed in Section 1.5 that a stochastic program of the form

$$\left.\begin{array}{ll} \text{"min"} & \mathbf{c}^T\mathbf{x} \\ \text{s.t.} & A\mathbf{x} = \mathbf{b} \\ & T(\tilde{\xi})\mathbf{x} = \mathbf{h}(\tilde{\xi}) \\ & \mathbf{x} \geq 0 \end{array}\right\} \tag{3.1}$$

is not a well-defined mathematical program. Ideally, we want to satisfy the constraints $T(\tilde{\xi})\mathbf{x} = \mathbf{h}(\tilde{\xi})$ but this cannot be done without knowledge of the realization of $\tilde{\xi}$. We therefore introduce the recourse variables $\mathbf{y}(\xi)$ such that the linear combination $W\mathbf{y}(\xi)$ of the recourse variables describ the violation $h(\xi) - T(\xi)x$ of these constraints. The penalty associated with the violation of the constraints is $q(\xi)^T y(\xi)$.

In other words, we have a set of first-stage decisions x to be taken without full information on some random events. Later, full information is received on the random events in the form of a realization ξ of a random vector. Then, second-stage or corrective actions (known as recourse actions) $y(\xi)$ are taken. For a given first-stage decision x, the second-stage decision y depends on the realization ξ. Formally, by setting

$$\begin{aligned} g_0(x) \; &: = c^T x \\ Q(x,\xi) \; &: = \min_{y} \{q(\xi)^T y \mid Wy = h(\xi) - T(\xi)x \,,\; y \geq 0\} \\ X \; &: = \{x \in \mathbb{R}^{n_1} \mid Ax = b \,,\, x \geq 0\} \end{aligned} \tag{3.2}$$

in (1.9), we obtain the general form of the *two-stage stochastic linear program with recourse*

$$\left.\begin{aligned} \min \quad & c^T x + E_{\tilde{\xi}}\left[Q(x,\tilde{\xi})\right] \\ \text{s.t.} \quad & Ax = b \\ & x \geq 0 \end{aligned}\right\} \tag{3.3}$$

where the *recourse function* is

$$Q(x,\xi) = \left\{ \begin{array}{ll} \min\limits_{y(\xi)} & q(\xi)^T y(\xi) \\ \text{s.t.} & Wy(\xi) = h(\xi) - T(\xi)x \\ & y(\xi) \geq 0 \end{array} \right] \tag{3.4}$$

and c and b are known vectors in $\mathbb{R}^{n_1}$ and $\mathbb{R}^{m_1}$ respectively, and A and W are known matrices of size $m_1 \times n_1$ and $m_2 \times n_2$ respectively. The matrix W is called the *recourse matrix* and we assume it is fixed, in order to yield properties of the feasibility set that are convenient for computation - see Section 3.3.1. For a given realization ξ, the problem data $q(\xi)$, $h(\xi)$ and $T(\xi)$ become known. For each ξ, $q(\xi)$ is $n_2 \times 1$, $h(\xi)$ is $m_2 \times 1$ and $T(\xi)$ is $m_2 \times n_1$. We call $T(\xi)$ the *technology matrix*. The first-stage decisions are represented by the $n_1 \times 1$ vector x and the second-stage decisions for a given realization ξ are represented by the $n_2 \times 1$ vector $y(\xi)$.

We also denote by $\Xi \subseteq \mathbb{R}^k$ the support of $\tilde{\xi}$, *i.e.* the smallest closed subset in $\mathbb{R}^k$ such that $P(\Xi) = 1$.

The dependence of $\mathbf{y}$ on ξ is completely different from the dependence of the other parameters on ξ. It is not a functional dependence but simply indicates that the decisions $\mathbf{y}$ are typically different under different realizations of $\tilde{\xi}$. Formally, the decisions $\mathbf{y}(\xi)$ are chosen such that the constraints in (3.4) are satisfied *almost surely*, *i.e.* with probability one.

A more condensed formulation of the problem is the deterministic equivalent

$$\left.\begin{array}{ll} \min & \mathbf{c}^T\mathbf{x} + \mathcal{Q}(\mathbf{x}) \\ \text{s.t.} & A\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \geq 0 \end{array}\right\} \tag{3.5}$$

where we define the *expected recourse function*

$$\mathcal{Q}(\mathbf{x}) = E_{\tilde{\xi}}\left[Q(\mathbf{x}, \xi)\right] \tag{3.6}$$

The deterministic equivalent (3.5) is known as the *implicit representation* of the recourse problem because the recourse function is represented implicitly. If the expected recourse function (3.6) can be expressed explicitly, then (3.5) has the form of an ordinary nonlinear program.

3.3 Feasibility Sets

3.3.1 Induced Constraints

In linear recourse problems, we adopt the convention that if a decision $\mathbf{x}$ leads to an infeasible second-stage problem (3.4) for any realization ξ (or set of realizations) with positive probability, then that decision is suboptimal. In other words, we reject any decision that could lead to an *undefined recourse action*, even if that decision induced an infinitely low cost function for other realizations. Thus for a decision $\mathbf{x}$ to be feasible, we require the second-stage program (3.4) to be feasible $\forall \xi \in \Xi$. (Strictly we only require this condition to hold almost surely and not necessarily $\forall \xi \in \Xi$.)

Depending on the recourse matrix W and the support Ξ, this is not necessarily true for all first-stage decisions $x \in X$, and it may become necessary to impose, in addition to $x \in X$, further restrictions on our first-stage decisions, called *induced constraints*. The *induced feasibility set* K is defined as

$$K := \{x \mid \mathcal{Q}(x) < \infty\} \tag{3.7a}$$

or

$$K := \{x \mid \forall \xi \in \Xi,\ \exists y(\xi) \geq 0 \text{ s.t. } Wy(\xi) = h(\xi) - T(\xi)x\} \tag{3.8}$$

provided that the two definitions coincide. This happens when $\tilde{\xi}$ has finite second moments - see Theorem 3 in Section 3.1(b) of Birge & Louveaux [10]. The first-stage decisions are restricted to $x \in X \cap K$, so that (3.5) may be redefined as

$$\left.\begin{array}{ll} \min & c^T x + \mathcal{Q}(x) \\ \text{s.t.} & x \in X \cap K \end{array}\right\} \tag{3.9}$$

An alternative definition of the induced feasibility set when $\tilde{\xi}$ has finite or countable support (i.e. when $\tilde{\xi}$ is discrete) is

$$K := \bigcap_{\xi \in \Xi} K(\xi) \tag{3.10}$$

where we define the *elementary feasibility set* for ξ as

$$K(\xi) := \{x \mid \exists y(\xi) \geq 0 \text{ s.t. } Wy(\xi) = h(\xi) - T(\xi)x\} = \{x \mid Q(x, \xi) < \infty\} \tag{3.11}$$

The following propositions show that K is a closed convex set and that K is a convex polyhedral set when the support is a finite set or a convex polyhedron.

Proposition 11 *When $\tilde{\xi}$ has a discrete distribution or a continuous distribution with finite second moments, the induced feasibility set K is closed and convex.*

Proof. Suppose that $\tilde{\xi}$ has a discrete distribution. Then for each ξ, the constraints $Wy(\xi) = h(\xi) - T(\xi)x$, $y(\xi) \geq 0$ are linear. Since the feasible set of any linear program is convex polyhedral and closed, the set in $\{x, y(\xi)\}$ that satisfies these constraints, and hence $K(\xi)$, is convex polyhedral and closed. Therefore the set $K = \bigcap_{\xi \in \Xi} K(\xi)$ is closed and convex since it is an intersection of closed convex sets. It is proved in Theorem 4(a) in Section 3.1(b) of Birge

& Louveaux [10] that when $\tilde{\xi}$ has a continuous distribution with finite second moments, K is closed and convex. ∎

Proposition 12 *If the support is a finite set* $\Xi = \{\xi^1, \ldots, \xi^r\}$, *then the induced feasibility set*

$$K = \{x \mid \forall j = 1, \ldots, r, \ \exists y(\xi^j) \geq 0 \text{ s.t. } Wy(\xi^j) = h(\xi^j) - T(\xi^j)x\} \qquad (3.12)$$

is a convex polyhedral set and hence $X \cap K$ *is a convex polyhedral set.*

Proof. From the proof of the previous proposition, each $K(\xi^j)$ is a convex polyhedral set. Therefore K is convex polyhedral since it is the intersection of a finite number of convex polyhedral sets. Also, X is convex polyhedral and therefore $X \cap K$ is convex polyhedral. ∎

Proposition 13 *If the support is a convex polyhedron* $\Xi = \text{conv}\{\xi^1, \ldots, \xi^r\}$, *and* $h(\xi)$ *and* $T(\xi)$ *are linear functions of the elements of* $\xi = (\xi_1, \ldots, \xi_k)^T$, *i.e.*

$$\begin{aligned} T(\xi) &= T^0 + \xi_1 T^1 + \ldots + \xi_k T^k \\ h(\xi) &= h^0 + \xi_1 h^1 + \ldots + \xi_k h^k \end{aligned}$$

where the T^i *are fixed* $m_2 \times n_1$ *matrices and the* h^i *are fixed* $m_2 \times 1$ *vectors, then the induced feasibility set* K *is a convex polyhedral set.*

Proof. We show that K corresponds to (3.12) and hence K is a convex polyhedral set by the previous proposition. Since the support Ξ is a convex polyhedron, it can be written as

$$\Xi = \text{conv}\{\xi^1, \ldots, \xi^r\} = \left\{\xi \mid \xi = \sum_{j=1}^{r} \lambda_j \xi^j, \ \sum_{j=1}^{r} \lambda_j = 1, \ \lambda_j \geq 0 \ \forall j\right\}$$

Consider the set $K' := \{x \mid \forall j = 1, \ldots, r, \ \exists y(\xi^j) \geq 0 \text{ s.t. } Wy(\xi^j) = h(\xi^j) - T(\xi^j)x\} \supseteq K$ where K is given by (3.8). Note that K' corresponds to (3.12). For any $x \in K'$ and for any $\xi \in \Xi$, it follows that

$$h(\xi) - T(\xi)x = h^0 + \xi_1 h^1 + \ldots + \xi_k h^k - T^0 x - \xi_1 T^1 x - \ldots - \xi_k T^k x$$

$$= \sum_{j=1}^{r} \lambda_j \left(\mathbf{h}^0 + \xi_1^j \mathbf{h}^1 + \ldots + \xi_k^j \mathbf{h}^k - T^0\mathbf{x} - \xi_1^j T^1 \mathbf{x} - \ldots - \xi_k^j T^k \mathbf{x} \right)$$

$$= \sum_{j=1}^{r} \lambda_j \left(\mathbf{h}(\xi^j) - T(\xi^j)\mathbf{x} \right) = \sum_{j=1}^{r} \lambda_j W \mathbf{y}(\xi^j) = W \sum_{j=1}^{r} \lambda_j \mathbf{y}(\xi^j)$$

Thus for $\mathbf{x} \in K'$ and for all $\xi \in \Xi$, there exists $\mathbf{y}(\xi) = \sum_{j=1}^{r} \lambda_j \mathbf{y}(\xi^j) \geq 0$ such that $W\mathbf{y}(\xi) = \mathbf{h}(\xi) - T(\xi)\mathbf{x}$. Therefore $\mathbf{x} \in K$ and $K' \subseteq K$ and hence $K' = K$. ■

In the previous section, we stated that the recourse matrix W was fixed. One of the reasons for this should now be apparent. If we allowed W to depend on $\boldsymbol{\xi}$, then $W(\xi)$ could have one or more rows of zeros for some realization ξ. In this case, it would be impossible to find a feasible $\mathbf{y}(\xi) \geq 0$ that satisfied $W\mathbf{y}(\xi) = \mathbf{h}(\xi) - T(\xi)\mathbf{x}$ if the appropriate elements of the right hand side were nonzero. This kind of difficulty rarely occurs for programs with a fixed recourse matrix and never occurs when the second moments of $\tilde{\boldsymbol{\xi}}$ are also finite. Another problem that can occur when the recourse matrix is not fixed or $\tilde{\boldsymbol{\xi}}$ does not have finite second moments is that the sets $K = \{\mathbf{x} \mid Q(\mathbf{x}, \xi) < \infty, \ \forall \xi \in \Xi\}$ and $\{\mathbf{x} \mid \mathcal{Q}(\mathbf{x}) < \infty\}$ might not coincide.

3.3.2 Relatively Complete Recourse

Computational advantages are obtained when the recourse matrix W has certain properties. One of these is the property of *relatively complete recourse* where every possible first-stage decision $\mathbf{x} \in X$ leads to a feasible second-stage problem. This implies that $X \subseteq K$ and hence we need only consider the first-stage feasibility set X. In other words, the problem of induced constraints does not exist.

Definition 18 (Relatively Complete Recourse) *A linear recourse program with a fixed recourse matrix W is said to have relatively complete recourse if*

$$\mathbf{h}(\xi) - T(\xi)\mathbf{x} \in \operatorname{pos} W, \ \forall \xi \in \Xi, \mathbf{x} \in X \tag{3.13}$$

where $\operatorname{pos} W$ *denotes the positive hull of the columns of W (see Appendix A.1).*

However, relatively complete recourse can be difficult to identify. An easier property to identify is that of *complete recourse*, where any possible $\mathbf{t} \in \mathbb{R}^{m_2}$ for the right hand side of the second-stage problem $\{W\mathbf{y} = \mathbf{t}\,,\ \mathbf{t} \geq 0\}$ leads to a feasible $\mathbf{y}$.

Definition 19 (Complete Recourse) *A linear recourse program with a fixed recourse matrix W is said to have complete recourse if* $\operatorname{pos} W = \mathbb{R}^{m_2}$.

Complete recourse implies that $K = \mathbb{R}^{m_2}$ and clearly it also implies relatively complete recourse. The following proposition enables us to recognize complete recourse matrices.

Proposition 14 *A fixed $m_2 \times n_2$ matrix W is a complete recourse matrix iff it has* $\operatorname{rank}(W) = m_2$ *and, assuming without loss of generality that its first m_2 columns $W_1, W_2, \ldots, W_{m_2}$ are linearly independent, the linear constraints*

$$\left.\begin{array}{l} W\mathbf{y} = 0 \\ y_i \geq 1\,,\ i = 1, \ldots, m_2 \\ \mathbf{y} \geq 0 \end{array}\right\} \tag{3.14}$$

have a feasible solution.

Proof. (Necessity) W is a complete recourse matrix iff $\{\mathbf{z} \mid \mathbf{z} = W\mathbf{y}\,,\ \mathbf{y} \geq 0\} = \mathbb{R}^{m_2}$. In order for W to form a basis in $\mathbb{R}^{m_2}$, it follows from standard results in linear algebra (see Anton [3], for example) that $\operatorname{rank}(W) = m_2$ must necessarily hold. Assume that W is a complete recourse matrix. For $\tilde{\mathbf{z}} = -\sum_{i=1}^{m_2} W_i \in \mathbb{R}^{m_2}$, the second-stage constraints $W\mathbf{y} = \tilde{\mathbf{z}}\,,\ \mathbf{y} \geq 0$ have a feasible solution $\tilde{\mathbf{y}} \geq 0$ such that

$$W\tilde{\mathbf{y}} = \sum_{i=1}^{m_2} W_i \tilde{y}_i + \sum_{i=m_2+1}^{n_2} W_i \tilde{y}_i = -\sum_{i=1}^{m_2} W_i = \tilde{\mathbf{z}}$$

Now if we define

$$y_i := \left\{ \begin{array}{ll} \tilde{y}_i + 1\,, & \text{for } i = 1, \ldots, m_2 \\ \tilde{y}_i\,, & \text{for } i = m_2 + 1, \ldots, n_2 \end{array} \right]$$

then

$$W\mathbf{y} = \sum_{i=1}^{m_2} W_i(\tilde{y}_i + 1) + \sum_{i=m_2+1}^{n_2} W_i \tilde{y}_i = W\tilde{\mathbf{y}} + \sum_{i=1}^{m_2} W_i = W\tilde{\mathbf{y}} - \tilde{\mathbf{z}} = 0$$

and $y_i \geq 1$, for $i = 1, \ldots, m_2$ so that $\mathbf{y} \geq \mathbf{0}$ satisfies (3.14), *i.e.* the constraints (3.14) are necessarily feasible.

(Sufficiency) Given that the constraints (3.14) have a feasible solution, say $\breve{y}$, we must show that a feasible solution $\mathbf{y} \geq 0$ exists to $W\mathbf{y} = \bar{\mathbf{z}}$, for an arbitrary $\bar{\mathbf{z}} \in \mathbb{R}^{m_2}$. Since the columns $W_1, \ldots, W_{m_2}$ are linearly independent, the system of linear equations $\sum_{i=1}^{m_2} W_i y_i = \bar{\mathbf{z}}$ has a unique solution $\bar{y}_1, \ldots, \bar{y}_{m_2}$. If $\bar{y}_i \geq 0$ for $i = 1, \ldots, m_2$ then $\bar{\mathbf{y}}$ is a feasible solution to $W\bar{\mathbf{y}} = \bar{\mathbf{z}}$ where we define $\bar{y}_i := 0$ for $i = m_2+1, \ldots, n_2$. Otherwise, we define $\gamma := \min\{\bar{y}_1, \ldots, \bar{y}_{m_2}\} < 0$ and

$$\hat{y}_i := \left\{ \begin{array}{ll} \bar{y}_i - \gamma \breve{y}_i & \text{for } i = 1, \ldots, m_2 \\ -\gamma \breve{y}_i & \text{for } i = m_2 + 1, \ldots, n_2 \end{array} \right]$$

Then

$$W\hat{\mathbf{y}} = \sum_{i=1}^{m_2} W_i (\bar{y}_i - \gamma \breve{y}_i) + \sum_{i=m_2+1}^{n_2} W_i (-\gamma \breve{y}_i) = \sum_{i=1}^{m_2} W_i \bar{y}_i - \gamma \sum_{i=1}^{n_2} W_i \breve{y}_i = \bar{\mathbf{z}} - \gamma \times 0$$

Now $\hat{y}_i \geq \bar{y}_i - \gamma \geq 0$ for $i = 1, \ldots, m_2$ since $-\gamma > 0$ and $\breve{y}_i \geq 1$ for $i = 1, \ldots, m_2$. Also, $\hat{y}_i \geq 0$ for $i > m_2$ since $\breve{y}_i \geq 0\ \forall i$. Therefore $\hat{\mathbf{y}} \geq 0$ is a feasible solution to $W\hat{\mathbf{y}} = \bar{\mathbf{z}}$. ■

3.4 Properties of the Recourse Function

Proposition 15 *For a stochastic program with fixed recourse, the second-stage function for a particular realization* ξ, $Q(\mathbf{x}, \xi) = \min\{\mathbf{q}^T\mathbf{y} \mid W\mathbf{y} = \mathbf{h} - T\mathbf{x}, \mathbf{x} \geq 0\}$, *is*

(a) a piecewise linear convex function in $\mathbf{x}$ *for all* $\mathbf{x} \in X \cap K$;

(b) a piecewise linear convex function in $(\mathbf{h}, T)$;

(c) a piecewise linear concave function in $\mathbf{q}$.

Proof. Consider two different vectors $\mathbf{x}_1, \mathbf{x}_2 \in X \cap K$ and let $\bar{\mathbf{x}} = \lambda\mathbf{x}_1 + (1-\lambda)\mathbf{x}_2$ where $\lambda \in (0,1)$. Note that $\bar{\mathbf{x}} \in X \cap K$ since $X \cap K$ is a convex set. Let $\mathbf{y}_1^*, \mathbf{y}_2^*$ and $\bar{\mathbf{y}}^*$ be optimal solutions of the associated problems, so that $Q(\mathbf{x}_1, \xi) = \mathbf{q}^T\mathbf{y}_1^*$, $Q(\mathbf{x}_2, \xi) = \mathbf{q}^T\mathbf{y}_2^*$ and $Q(\bar{\mathbf{x}}, \xi) = \mathbf{q}^T\bar{\mathbf{y}}^*$. Now $\lambda\mathbf{y}_1^* + (1-\lambda)\mathbf{y}_2^*$ is a feasible solution to $\min\{\mathbf{q}^T\mathbf{y} \mid W\mathbf{y} = \mathbf{h} - T\bar{\mathbf{x}}, \mathbf{y} \geq 0\}$ since $W(\lambda\mathbf{y}_1^* + (1-\lambda)\mathbf{y}_2^*) = \lambda W\mathbf{y}_1^* + (1-\lambda)W\mathbf{y}_2^* = \lambda(\mathbf{h} - T\mathbf{x}_1) + (1-\lambda)(\mathbf{h} - T\mathbf{x}_2) = \mathbf{h} - T\bar{\mathbf{x}}$ and hence it follows that

$$Q(\bar{\mathbf{x}}, \xi) \;=\; \mathbf{q}^T\bar{\mathbf{y}}^* \leq \mathbf{q}^T(\lambda\mathbf{y}_1^* + (1-\lambda)\mathbf{y}_2^*) = \lambda\mathbf{q}^T\mathbf{y}_1^* + (1-\lambda)\mathbf{q}^T\mathbf{y}_2^*$$

$$= \lambda Q(\mathbf{x}_1,\xi) + (1-\lambda)Q(\mathbf{x}_2,\xi)$$

and therefore $Q(\mathbf{x},\xi)$ is a convex function in $\mathbf{x}$ for fixed ξ.

Let $f(\mathbf{h},T) = \min_{\mathbf{y}} \{\mathbf{q}^T\mathbf{y} \mid W\mathbf{y} = \mathbf{h} - T\mathbf{x},\, \mathbf{y} \geq 0\}$ for fixed $\mathbf{q}$ and $\mathbf{x}$, *i.e.* $Q(\mathbf{x},\xi)$ viewed as a function of $\mathbf{h}$ and T (which in turn are functions of ξ). For different $(\mathbf{h}_1,T_1)$ and $(\mathbf{h}_2,T_2)$, let $(\overline{\mathbf{h}},\overline{T}) = \lambda(\mathbf{h}_1,T_1) + (1-\lambda)(\mathbf{h}_2,T_2)$ where $\lambda \in (0,1)$. Let $\mathbf{y}_1^*$, $\mathbf{y}_2^*$ and $\bar{\mathbf{y}}^*$ be optimal solutions of the associated problems, so that $f(\mathbf{h}_1,T_1) = \mathbf{q}^T\mathbf{y}_1^*$, $f(\mathbf{h}_2,T_2) = \mathbf{q}^T\mathbf{y}_2^*$ and $f(\overline{\mathbf{h}},\overline{T}) = \mathbf{q}^T\bar{\mathbf{y}}^*$. Since $\lambda\mathbf{y}_1^* + (1-\lambda)\mathbf{y}_2^*$ is a feasible solution to $\min\{\mathbf{q}^T\mathbf{y} \mid W\mathbf{y} = \overline{\mathbf{h}} - \overline{T}\mathbf{x},\, \mathbf{y} \geq 0\}$ it follows that

$$\begin{aligned} f(\overline{\mathbf{h}},\overline{T}) &= \mathbf{q}^T\bar{\mathbf{y}}^* \leq \mathbf{q}^T(\lambda\mathbf{y}_1^* + (1-\lambda)\mathbf{y}_2^*) = \lambda\mathbf{q}^T\mathbf{y}_1^* + (1-\lambda)\mathbf{q}^T\mathbf{y}_2^* \\ &= \lambda f(\mathbf{h}_1,T_1) + (1-\lambda) f(\mathbf{h}_2,T_2) \end{aligned}$$

and therefore $f(\mathbf{h},T)$ and hence $Q(\mathbf{x},\xi)$ are jointly convex in $(\mathbf{h},T)$.

Let $g(\mathbf{q}) = \min\{\mathbf{q}^T\mathbf{y} \mid W\mathbf{y} = \mathbf{h} - T\mathbf{x},\, \mathbf{y} \geq 0\}$ for fixed $\mathbf{h}, T$ and $\mathbf{x}$, *i.e.* $Q(\mathbf{x},\xi)$ viewed as a function of $\mathbf{q}$ (which is a function of ξ). For different $\mathbf{q}_1$ and $\mathbf{q}_2$, let $\overline{\mathbf{q}} = \lambda\mathbf{q}_1 + (1-\lambda)\mathbf{q}_2$ where $\lambda \in (0,1)$. Let $\mathbf{y}_1^*$, $\mathbf{y}_2^*$ and $\bar{\mathbf{y}}^*$ be optimal solutions of the associated problems, such that $g(\mathbf{q}_1) = \mathbf{q}_1^T\mathbf{y}_1^*$, $g(\mathbf{q}_2) = \mathbf{q}_2^T\mathbf{y}_2^*$ and $g(\overline{\mathbf{q}}) = \overline{\mathbf{q}}^T\bar{\mathbf{y}}^*$. Then

$$\begin{aligned} g(\overline{\mathbf{q}}) &= \overline{\mathbf{q}}^T\bar{\mathbf{y}}^* = (\lambda\mathbf{q}_1^T + (1-\lambda)\mathbf{q}_2^T)\,\bar{\mathbf{y}}^* = \lambda\mathbf{q}_1^T\bar{\mathbf{y}}^* + (1-\lambda)\mathbf{q}_2^T\bar{\mathbf{y}}^* \\ &\geq \lambda\mathbf{q}_1^T\mathbf{y}_1^* + (1-\lambda)\mathbf{q}_2^T\mathbf{y}_2^* = \lambda g(\mathbf{q}_1) + (1-\lambda)g(\mathbf{q}_2) \end{aligned}$$

and therefore $g(\mathbf{q})$ and hence $Q(\mathbf{x},\xi)$ are concave in $\mathbf{q}$.

The objective function $\mathbf{q}^T\mathbf{y}$ of $Q(\mathbf{x},\xi)$ varies linearly in $\mathbf{q}$ for each basis. Each linear piece or facet corresponds to a basis. The basis changes at the facet boundaries, and the objective function changes in a continuous manner at these boundaries. The function $Q(\mathbf{x},\xi)$ is therefore piecewise linear in $\mathbf{q}$. Alternatively, $Q(\mathbf{x},\xi)$ can be expressed in terms of the dual problem as $Q(\mathbf{x},\xi) = \max\{\pi^T(\mathbf{h} - T\mathbf{x}) \mid W^T\pi \leq \mathbf{q}\}$. It follows similarly that the objective is a linear function of $\mathbf{x}$ and $(\mathbf{h},T)$ for each basis and therefore $Q(\mathbf{x},\xi)$ is piecewise linear in $\mathbf{x}$ and $(\mathbf{h},T)$.
■

In the above proposition, $\mathbf{h}, T$ and $\mathbf{q}$ can be functions of ξ. The following corollaries (which follow immediately from the proposition) show that $Q(\mathbf{x}, \xi)$ is also piecewise linear and convex or concave in ξ when $\mathbf{h}$ and T or $\mathbf{q}$ are affine functions of ξ.

Corollary 2 *If $\mathbf{q}(\xi) \equiv \mathbf{q}_0$ is fixed and $T(\xi)$ and $\mathbf{h}(\xi)$ are affine functions of ξ, i.e. $T(\xi) = T_0 + \sum_i T^i \xi_i$ and $\mathbf{h}(\xi) = \mathbf{h}_0 + \sum_i \mathbf{h}^i \xi_i$, then $Q(\mathbf{x}, \xi)$ is piecewise linear and convex in ξ for fixed $\mathbf{x}$.*

Corollary 3 *If $\mathbf{h}$ and T are fixed and $\mathbf{q}(\xi)$ is an affine function of ξ, i.e. $\mathbf{h}(\xi) - T(\xi)\mathbf{x} \equiv \mathbf{h}_0 - T_0\mathbf{x}$ and $\mathbf{q}(\xi) \equiv \mathbf{q}_0 + \sum_i \mathbf{q}^i \xi_i$, then $Q(\mathbf{x}, \xi)$ is piecewise linear and concave in ξ for fixed $\mathbf{x}$.*

Proposition 16 *For a stochastic program with fixed recourse where $\tilde{\xi}$ has finite second moments,*

(a) $\mathcal{Q}(\mathbf{x})$ is a convex function and is finite on K.

(b) If $\tilde{\xi}$ has finite support, then $\mathcal{Q}(\mathbf{x})$ is piecewise linear.

(c) If $F(\xi)$ is continuous, then $\mathcal{Q}(\mathbf{x})$ is differentiable on K.

Proof. $\mathcal{Q}(\mathbf{x})$ is a convex function since it is a positive-weighted average of the functions $Q(\mathbf{x}, \xi)$ which are convex in $\mathbf{x}$ by the previous proposition. The finite second moments ensure convergence of the weighted average. The weighted average $\mathcal{Q}(\mathbf{x})$ is also finite on K since each $Q(\mathbf{x}, \xi)$ is finite on K by the definition of K. When $\tilde{\xi}$ has finite support, $\mathcal{Q}(\mathbf{x})$ is piecewise linear since it is a finite weighted average of the functions $Q(\mathbf{x}, \xi)$, each of which is piecewise linear in $\mathbf{x}$ by the previous proposition. For a proof of differentiability when $\tilde{\xi}$ has a continuous distribution, see Proposition 19 in Chapter 4 of this dissertation and Remark 1.2 in Section 1.4 of Kall & Wallace [28]. ■

In the following example, we calculate closed form expressions for a typical recourse function $\mathcal{Q}(x)$ in the case of a Normal distribution and a Poisson distribution, and illustrate that in these cases $\mathcal{Q}(x)$ has the properties asserted in the previous proposition.

Example

Let x represent the first-stage production of a given good. Let $\widetilde{\xi}$ be the demand for the same good. A typical second stage would consist of selling as much as possible, namely $\min(\xi, x)$ for a given realization ξ. If we were minimizing the objective, the expected recourse function would be $\mathcal{Q}(x) = -E_{\widetilde{\xi}}\left[\min\left(\widetilde{\xi}, x\right)\right]$.

1. If $\widetilde{\xi} \sim N\left(\mu, \sigma^2\right)$ then

$$\begin{aligned}\mathcal{Q}(x) &= -\int_{-\infty}^{x} \xi \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{1}{2}\left(\frac{\xi-\mu}{\sigma}\right)^2\right\} d\xi - \int_{x}^{\infty} x \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{1}{2}\left(\frac{\xi-\mu}{\sigma}\right)^2\right\} d\xi \\ &= \frac{\sigma}{\sqrt{2\pi}} \exp\left\{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right\} + (x-\mu)\Phi\left(\frac{x-\mu}{\sigma}\right) - x, \text{ for } x \geq 0 \qquad (3.15)\end{aligned}$$

The expected recourse function (3.15) is both convex and differentiable in x, since the density of a Normal distribution is continuous. Figure 3-1 illustrates $\mathcal{Q}(x)$ for a $N(\mu = 4, \sigma^2 = 4)$ distribution. Note that the objective function z is the sum of the first-stage objective cx (which is an increasing function of x for $c > 0$) and the expected recourse function $\mathcal{Q}(x)$ (which is a decreasing function of x). This means that the objective $z = cx + \mathcal{Q}(x)$ is not necessarily minimized by making x as large as possible, as one might be tempted to think whilst looking at the graph of the expected recourse function.

2. If $\widetilde{\xi} \sim P(\lambda)$ then

$$\mathcal{Q}(x) = -\sum_{j=0}^{\lfloor x \rfloor} j \frac{e^{-\lambda}\lambda^j}{j!} - \sum_{j=\lfloor x \rfloor+1}^{\infty} x \frac{e^{-\lambda}\lambda^j}{j!} = \sum_{j=0}^{\lfloor x \rfloor} (x-j) \frac{e^{-\lambda}\lambda^j}{j!} - x, \text{ for } x \geq 0 \qquad (3.16)$$

where $\lfloor x \rfloor$ denotes floor(x), *i.e.* the smallest integer less than or equal to x. The expected recourse function (3.16) is piecewise linear and convex in x, as we might expect since $\widetilde{\xi}$ has a discrete distribution with countable support rather than finite support. Figure 3-2 illustrates $\mathcal{Q}(x)$ for a $P(\lambda = 4)$ distribution. Note that this distribution has the same mean and variance (both of which are equal to 4) as the Normal distribution above.

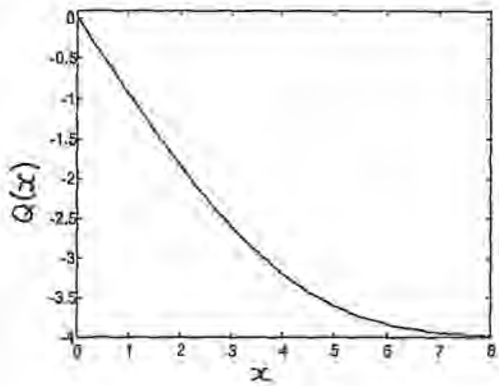

Figure 3-1: Recourse Function for a Normal ($\mu = 4, \sigma^2 = 4$) Distribution

3.5 Solution Methods

3.5.1 Difficulties with Continuous Random Variables

It was shown in the Proposition 16 of the previous section that when the random vector is characterized by a *continuous* distribution, the second-stage value function $\mathcal{Q}(\mathbf{x})$ is differentiable and convex, provided that the distribution has finite second moments and a continuous density function. This is not a restrictive condition since there are many continuous distributions that have these properties. Standard nonlinear programming techniques can then be applied to solve the recourse problem (3.5) if we can find an *analytic expression* for $\mathcal{Q}(\mathbf{x})$, such as the expressions that were found in our example. However, analytic expressions can only be found (without great effort) for small second-stage problems and for problems with a very specific structure.

In general, $\mathcal{Q}(\mathbf{x})$ can only be computed by *numerical integration* of $Q(\mathbf{x}, \boldsymbol{\xi})$ over $\boldsymbol{\xi}$ for a given value of $\mathbf{x}$. Most nonlinear programming methods also require the *gradient* of $\mathcal{Q}(\mathbf{x})$ which also involves numerical integration. Since numerical integration produces an effective computational method only when the random vector is of small dimensionality, the practical solution of stochastic programs having continuous random variables is, in general, a difficult problem.

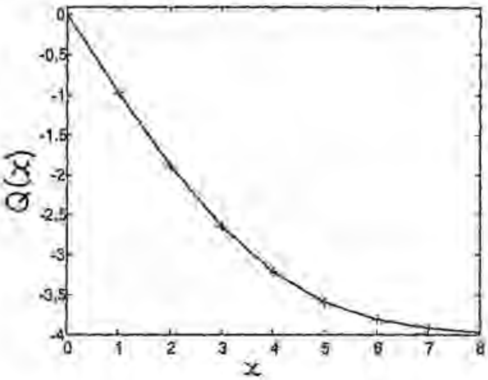

Figure 3-2: Recourse Function for a Poisson ($\lambda = 4$) Distribution

The most common approach is to *approximate* the continuous random variables by *discrete* ones, and to let the discretization become more and more refined, thus hoping that the solutions of the successive problems (with successively refined discrete distributions) will converge to the optimal solution of the original problem (with the continuous random variables). When the random variables have a finite distribution, $\mathcal{Q}(\mathbf{x})$ is piecewise linear and convex by Proposition 16 and the feasibility set $X \cap K$ is convex polyhedral by Proposition 12. Although $\mathcal{Q}(\mathbf{x})$ is computable as a finite sum, nonlinear programming methods that require gradients still cannot be used since $\mathcal{Q}(\mathbf{x})$ is no longer differentiable. However, efficient computational methods based on decomposition methods in linear programming have been developed to solve linear recourse programs with finite support.

3.5.2 Block Structure for Finite Distributions

When the random vector has a finite number r of realizations $\boldsymbol{\xi}^1, \ldots, \boldsymbol{\xi}^r$ with corresponding probabilities $p_1, \ldots, p_r$ where $\sum_{j=1}^{r} p_j = 1, \; p_j \geq 0 \; \forall j$, we show that the full deterministic equivalent linear program can always be formed. The expected recourse function $\mathcal{Q}(\mathbf{x})$ in (3.5) becomes

$$\mathcal{Q}(\mathbf{x}) = \sum_{j=1}^{r} p_j Q(\mathbf{x}, \boldsymbol{\xi}^j) \qquad (3.17)$$

where

$$Q(\mathbf{x}, \xi^j) = \min_{\mathbf{y}(\xi^j)} \{\mathbf{q}(\xi^j)^T \mathbf{y}(\xi^j) \mid W\mathbf{y}(\xi^j) = \mathbf{h}(\xi^j) - T(\xi^j)\mathbf{x}\,,\ \mathbf{y}(\xi^j) \geq 0\} \tag{3.18}$$

and the recourse problem (3.5) can be written in the following deterministic equivalent form, known as the *extensive form*,

$$\min\ \mathbf{c}^T\mathbf{x} + \sum_{j=1}^{r} p_j \mathbf{q}(\xi^j)^T \mathbf{y}(\xi^j)$$
$$\text{s.t.}\ \begin{cases} A\mathbf{x} = \mathbf{b} \\ T(\xi^j)\mathbf{x} + W\mathbf{y}(\xi^j) = \mathbf{h}(\xi^j)\,,\ j = 1, \ldots, r \\ \mathbf{x} \geq 0\,,\quad \mathbf{y}(\xi^j) \geq 0\,,\ j = 1, \ldots, r \end{cases} \tag{3.19}$$

provided that program (3.19) leads to the correct optimal values of $\mathbf{x}$ and $\mathbf{y}(\xi^j)$ $\forall j$. We now show that this is always the case. All the constraints involving $\mathbf{y}(\xi^j)$ are separate from all the constraints involving $\mathbf{y}(\xi^k)$ for $j \neq k$ and the terms in the objective are also separable in $\mathbf{y}(\xi^j)$ and therefore for a fixed first-stage decision $\mathbf{x} \in X$, (3.19) automatically leads to the correct optimal values $\mathbf{y}(\xi^j)$ of each second-stage program (3.18). Since we choose the optimal $\mathbf{x} \in X$, (3.19) produces the same optimal solution as (3.5) and hence the programs are always equivalent. Using the subscript j to denote dependence on the realization ξ^j, the extensive form (3.19) can be written

$$\min\ \mathbf{c}^T\mathbf{x} + \sum_{j=1}^{r} p_j \mathbf{q}_j^T \mathbf{y}_j$$
$$\text{s.t.}\ \begin{cases} A\mathbf{x} = \mathbf{b} \\ T_j\mathbf{x} + W\mathbf{y}_j = \mathbf{h}_j\,,\ j = 1, \ldots, r \\ \mathbf{x} \geq 0\,,\ \mathbf{y}_j \geq 0\,,\ j = 1, \ldots, r \end{cases} \tag{3.20}$$

The extensive form (3.20) is a linear program with the following special *block structure* that is called a *dual decomposition structure*.

Variable	x	y_1	y_2	$\cdots$	y_r	RHS
Objective	c^T	$p_1q_1^T$	$p_2q_2^T$	$\cdots$	$p_rq_r^T$	
1st stage	A					b
2nd stage ξ^1	T_1	W				h_1
2nd stage ξ^2	T_2		W			h_2
$\vdots$	$\vdots$			$\ddots$	$\vdots$	$\vdots$
2nd stage ξ^r	T_r			$\cdots$	W	h_r

(3.21)

In designing *algorithms* to solve stochastic programs with recourse, it has proved especially beneficial to take advantage of this structure and this has been the focus of much algorithmic work in stochastic programming. The structure (3.21) is amenable to *decomposition methods* such as the L-shaped method of Van Slyke & Wets [48] which is the subject of the next section. Other methods that exploit this structure include *inner linearization methods* and *basis factorization methods*.

3.5.3 The L-Shaped Method

In linear programming terms, the L-shaped method is a cutting plane technique that corresponds to a Dantzig-Wolfe decomposition [15] of the dual problem or a Benders decomposition [5] of the primal problem. The method extends these decomposition procedures by taking care of the induced feasibility constraints that must be satisfied in stochastic programming.

Consider formulation (3.5). The expected recourse function $\mathcal{Q}(x)$ in the objective is a nonlinear term that involves the solution of all the second-stage recourse programs. Since this term is clearly very expensive to compute, we aim to avoid evaluating it exactly. Furthermore, we must ensure that solutions lie within the induced feasibility set K, so that we are really solving (3.9).

It can easily be seen that the program

$$\left.\begin{array}{ll}\min\limits_{\mathbf{x},\theta} & \mathbf{c}^T\mathbf{x}+\theta \\ \text{s.t.} & \theta \geq \mathcal{Q}(\mathbf{x}) \\ & \mathbf{x} \in X \cap K\end{array}\right\} \tag{3.22}$$

produces the same optimal solution $\mathbf{x}^*$ as (3.9) with $\theta^* = \mathcal{Q}(\mathbf{x}^*)$, since θ should be chosen as its lower bound $\mathcal{Q}(\mathbf{x})$ when minimizing the objective. We aim to build a master linear program over a number of steps that is ultimately equivalent to (3.22) by sequentially adding constraints. The condition $\mathbf{x} \in X$ is immediately enforced by including the linear constraints $A\mathbf{x} = \mathbf{b}$, $\mathbf{x} \geq \mathbf{0}$. The conditions $\mathbf{x} \in K$ and $\theta \geq \mathcal{Q}(\mathbf{x})$ are however, more difficult to enforce. The condition $\mathbf{x} \in K$ is enforced by introducing a number of constraints known as *feasibility cuts*, while $\theta \geq \mathcal{Q}(\mathbf{x})$ is enforced by adding constraints that are linear approximations to $\mathcal{Q}$, called *optimality cuts*. At each step of the L-shaped method, either a feasibility cut or an optimality cut is added. It can be shown that the method either shows infeasibility or *converges* to a global optimum in a finite number of steps.

In the L-shaped method, we proceed by adding a constraint at each step so that our problem becomes increasingly constrained until ultimately we have all the constraints in place that are necessary to solve the recourse problem. Therefore, if the problem is infeasible at any intermediate stage, the original recourse problem is infeasible and we stop. For a given solution $\mathbf{x}^\nu$ of the νth master problem (defined below), we test to see if $\mathbf{x}^\nu$ leads to a feasible second-stage program for all realizations. If it does, we test for optimality, otherwise we add a feasibility cut and start again on the updated master problem. If $\mathbf{x}^\nu$ is optimal we stop, otherwise we add an optimality cut and return to the updated master problem. For a full development of the L-shaped algorithm, see Section 5.1 in Birge & Louveaux [10] or Section 3.2 in Kall & Wallace [28].

The L-shaped Algorithm

Step 0 Set $s = t = \nu = 0$.

Step 1 Set $\nu = \nu + 1$. Solve the master linear program (3.23) - (3.25).

$$\min \quad z = \mathbf{c}^T\mathbf{x} + \theta \tag{3.23}$$

$$\text{s.t. } A\mathbf{x} = \mathbf{b}$$

$$\gamma_\ell \mathbf{x} \geq \delta_\ell \,,\; \ell = 1, \ldots, s \tag{3.24}$$

$$\beta_\ell \mathbf{x} + \theta \geq \alpha_\ell \,,\; \ell = 1, \ldots, t \tag{3.25}$$

$$\mathbf{x} \geq 0 \,,\; \theta \in \mathbb{R}$$

The constraints (3.24) are called feasibility cuts and the constraints (3.25) are called optimality cuts. Let $(\mathbf{x}^\nu, \theta^\nu)$ be an optimal solution. If no optimality cuts are present, set $\theta^\nu = -\infty$ and do not include θ in the computation of $\mathbf{x}^\nu$.

Step 2 For $j = 1, \ldots, r$ solve the linear program

$$\left.\begin{array}{ll} \min & w_j = \mathbf{e}^T\mathbf{v}^+ + \mathbf{e}^T\mathbf{v}^- \\ \text{s.t.} & W\mathbf{y} + I_{m_2}\mathbf{v}^+ + I_{m_2}\mathbf{v}^- = \mathbf{h}_j - T_j\mathbf{x}^\nu \\ & \mathbf{y} \geq 0 \,,\; \mathbf{v}^+ \geq 0 \,,\; \mathbf{v}^- \geq 0 \end{array}\right\} \tag{3.26}$$

where $\mathbf{e}^T = (1, \ldots, 1)$, until for some j, the optimal value $w_j > 0$. In this case, let σ^ν be the associated $m_2 \times 1$ vector of simplex multipliers (see Appendix A.2.3). Define

$$\gamma_{s+1} = (\sigma^\nu)^T T_j \tag{3.27}$$

and

$$\delta_{s+1} = (\sigma^\nu)^T \mathbf{h}_j \tag{3.28}$$

and add the feasibility cut $\gamma_{s+1}\mathbf{x} \geq \delta_{s+1}$ to the constraint set (3.24). Set $s = s + 1$ and return to Step 1. If $w_j = 0$ for all j, go to Step 3.

Step 3 For $j = 1, \ldots, r$ solve the linear program

$$\left.\begin{array}{ll} \min & \mathbf{q}_j^T \mathbf{y} \\ \text{s.t.} & W\mathbf{y} = \mathbf{h}_j - T_j \mathbf{x}^\nu \\ & \mathbf{y} \geq 0 \end{array}\right\} \tag{3.29}$$

Let π_j^ν be the $m_2 \times 1$ vector of simplex multipliers associated with the optimal solution of the jth problem of type (3.29). Define

$$\beta_{t+1} = \sum_{j=1}^{r} p_j \left(\pi_j^\nu\right)^T T_j \tag{3.30}$$

and

$$\alpha_{t+1} = \sum_{j=1}^{r} p_j \left(\pi_j^\nu\right)^T \mathbf{h}_j \tag{3.31}$$

If $\theta^\nu \geq \alpha_{t+1} - \beta_{t+1}\mathbf{x}^\nu$ then stop, $\mathbf{x}^\nu$ is an optimal solution. Otherwise, set $t = t + 1$, add the optimality cut $\theta \geq \alpha_{t+1} - \beta_{t+1}\mathbf{x}$ to the constraint set (3.25), and return to Step 1.

3.5.4 Enhancements to the L-shaped Method

Step 2 of the L-shaped method determines whether a first-stage decision $\mathbf{x} \in X$ is also second-stage feasible, *i.e.* $\mathbf{x} \in K$, and involves introducing feasibility cuts. This step can be extremely time-consuming, as it may require the solution of up to r problems of the form (3.26). Moreover, this process may have to be repeated several times for different potential solutions. However, if the recourse problem has the property of complete recourse or relatively complete recourse as defined in Section 3.3.2, the second stage will always be feasible. Step 2 can then be omitted from the L-shaped method altogether, resulting in a large computational saving. This is a significant observation since many (or most!) well-modelled practical problems have relatively complete recourse. In some cases, a good understanding of the problem may make it possible to derive the necessary induced constraints beforehand and thus the feasibility step can be omitted from the L-shaped algorithm.

In Step 3 of the L-shaped method, all of the r second stage programs are solved in order to obtain their optimal simplex multipliers. These multipliers are then aggregated in (3.30) and

(3.31) to produce one optimality cut. Birge & Louveaux [9] proposed the *multicut version* of the L-shaped algorithm in which several optimality cuts (one per realization) are placed in each iteration of Step 3 instead of one. By sending disaggregate optimality cuts, more detailed information is given to the first stage. The number of returns to Step 1 is expected to decrease, although the master program grows rapidly as many more cuts are added. Whether the multicut version is more effective than the standard L-shaped method depends on the problem under consideration.

The L-shaped method involves the solution of many similar linear programs. For example, the programs may differ in the right-hand side and objective only and thus the same basis may produce solutions for several realizations. *Bunching* methods have been designed that avoid computational inefficiencies such as repetition in the calculations. These methods are useful in particular when the objective coefficients are deterministic, i.e. $q(\xi) \equiv q_0$. See Section 5.4 in Birge & Louveaux [10] for details on bunching methods.

3.5.5 Bounds and Approximations

Lower and upper bounds can be found on the expected value of the optimal objective function, i.e. the expected value of the wait-and-see solution. A straightforward application of this is that bounding methods can be used to bound the expected recourse function $\mathcal{Q}(x)$ in two-stage linear recourse programs. Using the Jensen lower bound and the Edmundson-Madansky upper bound or a piecewise linear upper bound, the expected recourse function can be approximated within any given tolerance. Such *approximation methods* are based on partitioning the support and refining the partitioning. See Sections 3.4 and 3.5 in Kall & Wallace [28] for details.

The L-shaped method can be adapted to utilize bounds and approximations. Instead of creating cuts that approximate $\mathcal{Q}(x)$, cuts are generated on a lower bound $\mathcal{L}(x)$, while the *stopping criterion* is provided by an upper bound $\mathcal{U}(x)$ and a tolerance ε.

These methods are useful mainly when there are *many random variables* and the deterministic equivalent problem becomes so large that even decomposition methods fail. Approximation methods calculate an approximate solution by replacing the exact deterministic equivalent problem by a smaller deterministic approximating problem through simplification of the distribution.

3.5.6 Simulation Methods

So far the methods that have been discussed, such as the L-shaped method, are deterministic methods. When these algorithms are repeated with the same input data, they will produce the same results each time. The opposite is true of *stochastic methods*, which usually do *not* produce the same results in two runs, even with the same input data. These methods have stopping criteria that are *statistical* in nature. Most of these methods operate on *samples* that are obtained by *Monte Carlo simulation*, rather than on all possible realizations of the distribution. One usually resorts to sampling methods (or bounding methods as discussed above) when the exact problem cannot be solved. The disadvantage of sampling approaches is that effort may be wasted on optimizing when the approximation is not accurate. This problem can be avoided to an extent by not optimizing completely.

The *stochastic decomposition method* of Higle & Sen [25] uses one stream of sample values to derive many cuts that eventually drop away as the number of iterations increases. A small number of sample values is generated before new cuts are added. The method only applies under relatively complete recourse.

3.5.7 Special Cases

The special structure of some stochastic programs provides computational advantages. One of the most important special structures in stochastic programming is *simple recourse*, which is discussed in Section 3.6. See the *production planning example* in Section 3.7 for another example where the structure of the problem is exploited to find an exact solution and to create and compare efficient solution techniques.

Network problems are specially structured linear programs. Stochastic network problems arise when there are random elements in the network. Efficient solution methods exist for deterministic networks and similarly, efficient methods have been devised to solve stochastic networks - see Chapter 6 in Kall & Wallace [28].

3.5.8 Methods for Multistage Programs

Although the multistage stochastic linear program with a finite number of possible future realizations has a deterministic equivalent linear program (see Section 3.5 in Birge & Louveaux [10]), the structure of this problem is more complex than that of the two-stage problem. These deterministic equivalents tend to expand in size extremely quickly. Solution methods for two-stage methods can be generalized to the multistage case but result in additional complications and can be difficult to implement. *Nested decomposition methods*, such as the nested L-shaped method for multistage stochastic linear programs (see Section 7.1 in Birge & Louveaux [10]), have been implemented successfully by Gassmann [23], for example.

3.6 Simple Recourse

A *simple recourse* problem is a linear recourse program of the form (3.3) where

$$\left.\begin{aligned} W &= (I, -I) \\ T(\xi) &\equiv T \\ h(\xi) &\equiv \xi \end{aligned}\right\} \tag{3.32}$$

in the second-stage program (3.4). To correspond with W we partition $\mathbf{y}$ and $\mathbf{q}$ as

$$\mathbf{y} = \begin{pmatrix} \mathbf{y}^+ \\ \mathbf{y}^- \end{pmatrix}, \mathbf{q} = \begin{pmatrix} \mathbf{q}^+ \\ \mathbf{q}^- \end{pmatrix}$$

In other words, in simple recourse we assume that the technology matrix T and objective coefficients $\mathbf{q}$ are fixed and that there is randomness in the right-hand side only. Furthermore, W is a complete recourse matrix. The recourse variables $\mathbf{y}^+$ represent the positive elements of the right-hand side $\boldsymbol{\xi} - T\mathbf{x}$, while the recourse variables $\mathbf{y}^-$ represent the negative elements of the right-hand side. Thus the recourse variables (and hence the penalties) are defined for every possible value of the right-hand side. By duality in linear programming (see Appendix A.2.3), we have for the recourse function

$$Q(\mathbf{x}, \xi) \;=\; \min_{y} \{ \mathbf{q}^{+T}\mathbf{y}^+ + \mathbf{q}^{-T}\mathbf{y}^- \mid y^+ + y^- = \xi - T\mathbf{x}, y^+ \geq 0, y^- \geq 0 \}$$

$$= \max_{\pi} \left\{ (\xi - T\mathbf{x})^T \pi \mid -\mathbf{q}^- \le \pi \le \mathbf{q}^+ \right\} \tag{3.33}$$

The dual formulation (3.33) clearly has a solution if and only if $\mathbf{q}^+ \ge -\mathbf{q}^-$ and therefore we make the necessary assumption that $\mathbf{q}^+ + \mathbf{q}^- \ge 0$ which is equivalent to solvability of the second-stage problem. If we define

$$\chi := T\mathbf{x} \tag{3.34}$$

and $\chi_i := T_i\mathbf{x}$ as the ith element of χ (where T_i is the ith row of T), then the solution π^* of the dual

$$\begin{aligned} Q(\mathbf{x}, \xi) &= \max \left\{ (\xi - \chi)^T \pi \mid -\mathbf{q}^- \le \pi \le \mathbf{q}^+ \right\} \\ &= \max \left\{ \sum_{i=1}^{m_2} (\xi_i - \chi_i)^T \pi_i \mid -q_i^- \le \pi_i \le q_i^+ \, , \, i = 1, \dots, m_2 \right\} \end{aligned} \tag{3.35}$$

is clearly

$$\pi_i^* = \left\{ \begin{array}{ll} q_i^+ & \text{if } \xi_i - \chi_i > 0 \\ -q_i^- & \text{if } \xi_i - \chi_i < 0 \end{array} \right] \tag{3.36}$$

The equality case $\xi_i - \chi_i = 0$ is not important since it adds zero to the summation and we can choose any $\pi_i^* \in [-q_i^-, q_i^+]$. By convention we choose $\pi_i^* = -q_i^-$. Therefore

$$Q(\mathbf{x}, \xi) = \sum_{i=1}^{m_2} Q_i(\chi_i, \xi_i)$$

where

$$Q_i(\chi_i, \xi_i) = \left\{ \begin{array}{ll} (\xi_i - \chi_i)\, q_i^+ & \text{if } \chi_i < \xi_i \\ -(\xi_i - \chi_i)\, q_i^- & \text{if } \chi_i \ge \xi_i \end{array} \right] \tag{3.37}$$

The expected recourse function is

$$\begin{aligned} \mathcal{Q}(\mathbf{x}) &= E_{\tilde{\xi}} \left[Q(\mathbf{x}, \tilde{\xi}) \right] = \int_{\Xi} Q(\mathbf{x}, \xi) f_{\tilde{\xi}}(\xi) d\xi = \sum_{i=1}^{m_2} \int_{\Xi} Q_i(\chi_i, \xi_i) f_{\tilde{\xi}}(\xi) d\xi \\ &= \sum_{i=1}^{m_2} \left\{ q_i^+ \int_{\xi_i > \chi_i} (\xi_i - \chi_i)\, f_{\tilde{\xi}}(\xi) d\xi - q_i^- \int_{\xi_i \le \chi_i} (\xi_i - \chi_i)\, f_{\tilde{\xi}}(\xi) d\xi \right\} \\ &= \sum_{i=1}^{m_2} \mathcal{Q}_i(\chi_i) \end{aligned} \tag{3.38}$$

where

$$\mathcal{Q}_i(\chi_i) = q_i^+ \int_{\chi_i}^{\infty} (\xi_i - \chi_i) f_i(\xi_i) d\xi_i - q_i^- \int_{-\infty}^{\chi_i} (\xi_i - \chi_i) f_i(\xi_i) d\xi_i \tag{3.39}$$

Expression (3.39) shows that the marginal distributions $f_i(\cdot)$ of each of the right-hand side elements $\tilde{\xi}_i$ are sufficient to evaluate the expected recourse function, since the other variables (*i.e.* all the elements of $\tilde{\xi}$ except for $\tilde{\xi}_i$) are integrated out over their support. Furthermore, (3.38) shows that $\mathcal{Q}(\mathbf{x})$ is a separable function in the χ_i.

These properties of simple recourse can be extremely useful in developing efficient computational methods, or even in solving the problem exactly. The property of *separability* is particularly useful in evaluating the multiple integral or multiple summation when calculating $\mathcal{Q}(\mathbf{x})$. For simple recourse problems with continuous distributions, problem (3.5) can be solved by nonlinear programming methods, where $\mathcal{Q}(\mathbf{x})$ is treated explicitly as a nonlinear function, or as a sum of nonlinear functions according to (3.38) and (3.39). See Section 3.6 in Kall & Wallace [28] and Sections 5.7 and 6.4 in Birge & Louveaux [10] for further development of methods pertaining to simple recourse.

3.7 Production Planning Example

The aim of this section is to illustrate the concepts of stochastic linear programming with recourse by means of an example, which was taken from Section 1.2 of Kall & Wallace [28]. Kall & Wallace solved the problem by approximating the distribution using a questionable simulation procedure, but gave no indication of the accuracy of their solution. In this section, the structure of the recourse problem is exploited to find the exact solution by analytic methods and to devise efficient methods of solution by numerical integration and by discretization. The solution methods are compared with each other and it is shown that the solutions appear to converge towards the exact solution, while the method of Kall & Wallace yields inaccurate results. The results also provide some insight into the accuracy of the solution obtained from a given level of refinement of the approximating distribution. Deterministic models for the problem are solved and compared with the stochastic model, and the value of the stochastic solution and the expected value of perfect information are calculated. The wait-and-see solutions and the average wait-and-see solution are also investigated. Chance-constrained models with joint chance constraints and separate chance constraints are formulated and compared with the recourse model.

3.7.1 An Oil Refinery's Production Problem

An oil refinery wants to decide the cheapest production plan for the forthcoming week. The producer must decide how many barrels x_i of crude oil to purchase at the beginning of the week from country $i = 1, 2$. A total of at most 100 barrels can be purchased. The costs of the barrels are \$20 and \$30 per unit respectively. Two different products - petrol and fuel oil - are produced simultaneously from the crude oil. The weekly demand of the refinery's customers for the two products varies randomly and can be modelled as $180 + \tilde{\xi}_1$ and $162 + \tilde{\xi}_2$ units of petrol and fuel oil respectively, where $\tilde{\xi}_1 \sim N(0, 144)$ and $\tilde{\xi}_2 \sim N(0, 81)$ independently. The customers expect their actual demands for the week to be met, even though the producer does not know in advance what it is going to be. Due to the nature of the production process and the quality of the crude oil, each barrel of crude oil from the first country always produces 3 units of fuel oil and each barrel of crude oil from the second country always produces 6 units of petrol. However, the other productivities vary randomly as follows. Each barrel of crude oil from the first country produces $2 + \tilde{\xi}_3$ units of petrol and each barrel of crude oil from the second country produces $3.4 - \tilde{\xi}_4$ units of fuel oil, where $\tilde{\xi}_3 \sim U(-0.8\,, 0.8)$ and $\tilde{\xi}_4 \sim Exp(2.5)$ independently of each other and of $\tilde{\xi}_1$ and $\tilde{\xi}_2$. The actual productivities are only observed during the production process itself. The production plan must be decided at the beginning of the week and cannot be changed during the week. The producer is thus faced with the problem

$$\left.\begin{array}{ll} \text{"min"} & 2x_1 + 3x_2 \\ \text{s.t.} & x_1 + x_2 \leq 100 \\ & (2 + \tilde{\xi}_3)x_1 + 6x_2 \geq 180 + \tilde{\xi}_1 \\ & 3x_1 + (3.4 - \tilde{\xi}_4)x_2 \geq 162 + \tilde{\xi}_2 \\ & x_1 \geq 0\,,\ x_2 \geq 0 \end{array}\right\} \tag{3.40}$$

where the objective (the cost function) is in units of \$10, and where $\tilde{\xi}_1 \sim N(0, 144)$, $\tilde{\xi}_2 \sim N(0, 81)$, $\tilde{\xi}_3 \sim U(-0.8, 0.8)$ and $\tilde{\xi}_4 \sim Exp(2.5)$, all independently. However, this is not a well-defined mathematical program, as was explained in Section 1.5.

3.7.2 Deterministic Formulation

Since $E(\widetilde{\xi}_1) = E(\widetilde{\xi}_2) = E(\widetilde{\xi}_3) = 0$ and $E(\widetilde{\xi}_4) = 0.4$, the expected value problem is

$$\left.\begin{array}{ll} \min & z = 2x_1 + 3x_2 \\ \text{s.t.} & x_1 + x_2 \leq 100 \\ & 2x_1 + 6x_2 \geq 180 \\ & 3x_1 + 3x_2 \geq 162 \\ & x_1 \geq 0\,, x_2 \geq 0 \end{array}\right\} \tag{3.41}$$

with the optimal solution

$$\left.\begin{array}{r} x_1^* = 36 \\ x_2^* = 18 \\ z^* = 126 \end{array}\right\} \tag{3.42}$$

This would be the optimal production plan if no variability was inherent in the problem.

3.7.3 Fat Solution

We can obtain confidence intervals for the unknown parameters as

$$\begin{array}{ll} \xi_1 \in [-30.909952, 30.909952] & \text{2-sided 99\% C.I.} \\ \xi_2 \in [-23.182464, 23.182464] & \text{2-sided 99\% C.I.} \\ \xi_3 \in [-0.8, 0.8] & \text{100\% C.I.} \\ \xi_4 \in [0, 1.842068] & \text{1-sided 99\% C.I.} \end{array} \tag{3.43}$$

The producer may want to look for a *safe* or *conservative* production plan, *i.e.* one that would be feasible for nearly all possible realizations of the parameters. Such a production plan is called a *fat solution* (refer back to Section 1.8) and reflects total risk aversion of the decision maker. The fat solution is determined by assuming *pessimistic* realized values for all of the parameters which, in this example, is equivalent to assuming high demands and low productivities. This corresponds to choosing ξ_1, ξ_2 and ξ_4 at their upper limits and ξ_3 at its lower limit. The linear program then becomes

$$\left.\begin{array}{ll}\min & z = 2x_1 + 3x_2 \\ \text{s.t.} & x_1 + x_2 \leq 100 \\ & 1.2x_1 + 6x_2 \geq 210.909952 \\ & 3x_1 + 1.557932x_2 \geq 185.182464 \\ & x_1 \geq 0\,,\, x_2 \geq 0\end{array}\right\} \tag{3.44}$$

which yields the solution

$$\left.\begin{array}{l}x_1^* = 48.511349 \\ x_2^* = 25.449389 \\ z^* = 173.370865\end{array}\right\} \tag{3.45}$$

The objective value $z^* \simeq 173$ is far more expensive than the objective $z^* = 126$ of the expected value problem. A problem with fat solutions in general is that they are rather *expensive* in the first stage. However, they are usually very cheap in the second stage, as we will see in Section 3.7.11.

3.7.4 Recourse Formulation

It may well be possible for the producer to find a cheaper production plan and still meet the demands of the clients if it is possible to set up an emergency (overtime) production plan or to buy the shortage in the market. Suppose that the products can be purchased in the market at a price of 7 and 12 per unit respectively. We can then formulate the second-stage program for given decisions x_1 and x_2 and realization $\boldsymbol{\xi}$ as

$$\left.\begin{array}{ll}\min & 7y_1(\xi) + 12y_2(\xi) \\ \text{s.t.} & (2 + \xi_3)x_1 + 6x_2 + y_1(\xi) \geq 180 + \xi_1 \\ & 3x_1 + (3.4 - \xi_4)x_2 + y_2(\xi) \geq 162 + \xi_2 \\ & y_1(\xi) \geq 0\,,\, y_2(\xi) \geq 0\end{array}\right\} \tag{3.46}$$

which immediately yields the optimal solution

$$\left.\begin{aligned} y_1^*(\xi) &= \max(180 + \xi_1 - (2 + \xi_3)x_1 - 6x_2\,,\, 0) \\ y_2^*(\xi) &= \max(162 + \xi_2 - 3x_1 - (3.4 - \xi_4)x_2\,,\, 0) \end{aligned}\right\} \tag{3.47}$$

Hence we get the two-stage recourse program

$$\begin{aligned} z(x_1, x_2) = \min \quad & 2x_1 + 3x_2 + \mathcal{Q}(x_1, x_2) \\ \text{s.t.} \quad & \begin{cases} x_1 + x_2 \le 100 \\ x_1 \ge 0\,,\, x_2 \ge 0 \end{cases} \end{aligned} \tag{3.48}$$

where

$$\mathcal{Q}(x_1, x_2) = E_{\tilde{\xi}}\,[7y_1^*(\xi) + 12\,y_2^*(\xi)]$$

and $y_1^*(\xi)$ and $y_2^*(\xi)$ are given by (3.47). The recourse problem (3.48) involves continuous distributions and can be solved by any of the following methods:

1. Exact solution by analytic methods.
2. Accurate calculation of $\mathcal{Q}(x_1, x_2)$ by numerical integration, and accurate optimization by nonlinear programming methods.
3. Approximate calculation of $\mathcal{Q}(x_1, x_2)$ by discretization and summation, and accurate optimization of the approximate problem by nonlinear programming methods.
4. Discretization of the distribution and exact optimization of the approximate problem by the L-shaped method.

In Sections 3.7.5 - 3.7.7, the recourse problem (3.48) is solved by these methods, and the results and computational effort are compared. Methods 3 and 4 produce the same solution for the same discretization, provided that the nonlinear methods are sufficiently accurate.

3.7.5 Exact Solution

The expected recourse function can be written as

$$\mathcal{Q}(x_1, x_2) = \mathcal{Q}_1(x_1, x_2) + \mathcal{Q}_2(x_1, x_2) \tag{3.49}$$

where

$$Q_1(x_1,x_2) = 7\int_{-0.8}^{0.8}\int_{6x_2-180+(2+\xi_3)x_1}^{\infty}(180+\xi_1-(2+\xi_3)x_1-6x_2)\frac{e^{-\frac{\xi_1^2}{288}}}{12\sqrt{2\pi}}\cdot\frac{1}{1.6}d\xi_1 d\xi_3$$

and

$$Q_2(x_1,x_2) = 12\int_{0}^{\infty}\int_{3x_1-162+(3.4-\xi_4)x_2}^{\infty}(162+\xi_2-3x_1-(3.4-\xi_4)x_2)\frac{e^{-\frac{\xi_2^2}{162}}}{9\sqrt{2\pi}}\cdot 2.5e^{-2.5\xi_4}d\xi_2\, d\xi_4$$

We solve (3.48) as an unconstrained problem and hope that the constraints will be satisfied. At an unconstrained optimum, the conditions

$$\left.\begin{array}{l}\frac{\partial z}{\partial x_1} = 2+\frac{\partial Q_1}{\partial x_1}+\frac{\partial Q_2}{\partial x_1} = 0\\ \frac{\partial z}{\partial x_2} = 3+\frac{\partial Q_1}{\partial x_2}+\frac{\partial Q_2}{\partial x_2} = 0\end{array}\right\} \tag{3.50}$$

must be satisfied. Using Leibnitz's rule for differentiation under the integral sign, and evaluating the resulting integrals, we obtain the partial derivatives

$$\begin{aligned}\frac{\partial Q_1}{\partial x_1} &= 7\int_{-0.8}^{0.8}\int_{6x_2-180+(2+\xi_3)x_1}^{\infty}\frac{-5\sqrt{2}}{192\sqrt{\pi}}(2+\xi_3)\exp\left[-\frac{\xi_1^2}{288}\right]d\xi_1\, d\xi_3\\ &= -14+\frac{26.25}{\sqrt{2\pi}x_1^2}(-6x_2+180+2.8x_1)\exp\left[-\frac{(6x_2-180+2.8x_1)^2}{288}\right]\\ &\quad -\frac{26.25}{\sqrt{2\pi}x_1^2}(-6x_2+180+1.2x_1)\exp\left[-\frac{(6x_2-180+1.2x_1)^2}{288}\right]\\ &\quad +\frac{35}{8x_1^2}(-16272+3.92x_1^2+1080x_2-18x_2^2)\,\Phi\left(\frac{6x_2-180+2.8x_1}{12}\right)\\ &\quad -\frac{35}{8x_1^2}(-16272+0.72x_1^2+1080x_2-18x_2^2)\,\Phi\left(\frac{6x_2-180+1.2x_1}{12}\right)\end{aligned}$$

$$\begin{aligned}\frac{\partial Q_1}{\partial x_2} &= 7\int_{-0.8}^{0.8}\int_{6x_2-180+(2+\xi_3)x_1}^{\infty}\frac{-5\sqrt{2}}{32\sqrt{\pi}}\exp\left[-\frac{\xi_1^2}{288}\right]d\xi_1 d\xi_3\\ &= -42+\frac{315}{\sqrt{2\pi}x_1}\left(\exp\left[-\frac{(6x_2-180+2.8x_1)^2}{288}\right]-\exp\left[-\frac{(6x_2-180+1.2x_1)^2}{288}\right]\right)\\ &\quad +26.25\left(2.8+\frac{6x_2-180}{x_1}\right)\Phi\left(\frac{6x_2-180+2.8x_1}{12}\right)\end{aligned}$$

$$-26.25\left(1.2+\frac{6x_2-180}{x_1}\right)\Phi\left(\frac{6x_2-180+1.2x_1}{12}\right)$$

$$\begin{aligned}
\frac{\partial Q_2}{\partial x_1} &= 12\int_0^\infty\int_{3x_1-162+(3.4-\xi_4)x_2}^\infty \frac{-5\sqrt{2}}{12\sqrt{\pi}}\exp\left[-\frac{\xi_2^2}{162}-2.5\xi_4\right]d\xi_2\,d\xi_4\\
&= -36+36\Phi\left(\frac{3x_1-162+3.4x_2}{9}\right)\\
&\quad -36\exp\left[\frac{253.125}{x_2}-7.5x_1+405-8.5x_2\right]\Phi\left(\frac{3x_1-162+3.4x_2}{9}-\frac{45}{2x_2}\right)
\end{aligned}$$

$$\begin{aligned}
\frac{\partial Q_2}{\partial x_2} &= 12\int_0^\infty\int_{3x_1-162+(3.4-\xi_4)x_2}^\infty \frac{5\sqrt{2}}{36\sqrt{\pi}}(-3.4+\xi_4)\exp\left[-\frac{\xi_2^2}{162}-2.5\xi_4\right]d\xi_2\,d\xi_4\\
&= -36+36\Phi\left(\frac{3x_1-162+3.4x_2}{9}\right)+\frac{108}{\sqrt{2\pi}x_2}\exp\left[-\frac{(3x_1-162+3.4x_2)^2}{162}\right]\\
&\quad +\frac{12}{x_2^2}\left(-202.5-162x_2+3x_1x_2+0.4x_2^2\right)\exp\left[\frac{1}{x_2}\left(\frac{253.125}{x_2}-7.5x_1+405-8.5x_2\right)\right]\\
&\quad \times\Phi\left(\frac{3x_1-162+3.4x_2}{9}-\frac{45}{2x_2}\right)
\end{aligned}$$

See Appendix B.3 for the integrals that were used in these calculations. Using MAPLE (see Appendix A.5) to solve (3.50) yields the optimal solution

$$\left.\begin{aligned}
x_1^* &= 39.1712930\\
x_2^* &= 22.8577789\\
RP = z\left(x_1^*, x_2^*\right) = z^* &= 153.8547495
\end{aligned}\right\} \qquad (3.51)$$

The expected value of the deterministic solution under the stochastic model is

$$EEV = z(36,18) = 239.0644924$$

so that the value of the stochastic solution is

$$VSS = EEV - RP = 239.0644924 - 153.8547495 = 85.2097429$$

Table 3.1: Numerical Results using Numerical Integration

ε_1	ε_2	x_1	x_2	z
10^{-2}	10^{-2}	39.1917	22.8576	153.854204
10^{-4}	10^{-2}	39.181533	22.855198	153.853623
10^{-3}	10^{-3}	39.178710	22.848018	153.853758
10^{-4}	10^{-4}	39.179642	22.847782	153.853599
10^{-5}	10^{-5}	39.170322	22.858769	153.853607
10^{-6}	10^{-6}	39.170327	22.858765	153.853607

which corresponds to a reduction in expected cost of 85.2097429 / 239.0644924 = 35.6%.

In practice, this approach of finding the exact solution would hardly ever be used because an excessive amount of work is involved (especially in evaluating the integrals analytically), even in a small problem such as this. In other problems, the integrals may well be impossible to evaluate analytically.

3.7.6 Solution by Numerical Integration

As shown in (3.49), the expected recourse function can be written as the sum of two double integrals. These integrals can be calculated numerically, and therefore (3.48) can be solved by nonlinear programming methods. The accuracy of the optimal solution obtained by this method will depend on both the tolerance ε_1 of the numerical integration procedure and the tolerance ε_2 of x_1 and x_2 in the optimization procedure. As $\varepsilon_1 \to 0$ and $\varepsilon_2 \to 0$, we expect the solution to approach the exact optimal solution (3.51). Table 3.1 gives numerical results that were obtained using MATLAB [1] for different tolerances. The numerical solution is close to the analytical one, especially when the tolerances are small. However, it becomes increasingly difficult to get answers for smaller tolerances. As a rough observation, I found that the computing time changed by orders of magnitude as the tolerances changed by orders of magnitude. Also, it appears that only limited accuracy can be obtained by the particular numerical integration procedure that was used (especially when integrating over an infinite range), and hence the accuracy of the optimization procedure is limited.

[1] The procedure QUAD2DG from the Numerical Integration Toolbox was used. It implements a *Gaussian quadrature* integration scheme.

3.7.7 Solution by Discretization

If the random vector $\tilde{\xi}$ is discretized into S possible realizations ξ^s with associated probabilities $p_s = \Pr[\xi = \xi^s]$ for $s = 1, \ldots, S$, where $\sum_{s=1}^{S} p_s = 1$, then the recourse problem (3.48) can be written as the linear program

$$\left.\begin{array}{ll} \min & 2x_1 + 3x_2 + \sum_{s=1}^{S} p_s\{7y_1(\xi^s) + 12y_2(\xi^s)\} \\ \text{s.t.} & x_1 + x_2 \leq 100 \\ & (2 + \xi_3^s)x_1 + 6x_2 + y_1(\xi^s) \geq 180 + \xi_1^s, \ \forall s \\ & 3x_1 + (3.4 - \xi_4^s)x_2 + y_2(\xi^s) \geq 162 + \xi_2^s, \ \forall s \\ & x_1 \geq 0, \ x_2 \geq 0 \\ & y_1(\xi^s) \geq 0, \ y_2(\xi^s) \geq 0, \ \forall s \end{array}\right\} \tag{3.52}$$

This linear program has the block structure (3.21) and can be solved by ' shaped method. However, since the second-stage program is always feasible with the solı.

$$\left.\begin{array}{l} y_1^*(\xi^s) = y_1^*(\xi_1^s, \xi_3^s) = \max(180 + \xi_1^s - (2 + \xi_3^s)x_1 - 6x_2, 0) \\ y_2^*(\xi^s) = y_2^*(\xi_2^s, \xi_4^s) = \max(162 + \xi_2^s - 3x_1 - (3.4 - \xi_4^s)x_2, 0) \end{array}\right\} \tag{3.53}$$

we can apply the L-shaped method without Step 2, as was discussed in Section 3.5.4. An alternative method is to exploit the separability of the expected recourse function $\mathcal{Q}(x_1, x_2)$ in (ξ_1, ξ_3) and (ξ_2, ξ_4) to calculate it explicitly for a given x_1 and x_2. Note that the value of $\mathcal{Q}(x_1, x_2)$ calculated in this away is only approximate because the discretized distributions approximate the continuous distributions. $\mathcal{Q}(x_1, x_2)$ can thus be approximated as

$$\mathcal{Q}(x_1, x_2) = \sum_{s=1}^{S} 7p_s y_1^*(\xi_1^s, \xi_3^s) + \sum_{s=1}^{S} 12p_s y_2^*(\xi_2^s, \xi_4^s) \tag{3.54}$$

The recourse problem (3.48) can then be solved within a given to ɪrance by nonlinear programming methods. Since the random variables are indr pendent, they can be discretized separately, so that the realizations of the random vector are obtained by rı nning through all possible realizations of each variable with the others fixed. I compare the results that were obtained using two different methods of discretization.

Table 3.2: Numerical Results for Discretization by Simulation

S_1	S_2	S_3	S_4	S	K	x_1	x_2	z
5	7	9	11	3465	10,000	37.610033	23.495563	150.629475
15	15	15	15	50625	150,000	37.763390	23.469839	150.968494
100	100	100	100	10^8	200,000	38.152242	23.118031	151.023148

Discretization by Simulation

In Section 1.3 of [28], Kall & Wallace describe a method to discretize the random variables by simulation. Their method involves truncating the distributions at their upper and lower confidence limits. Any simulated values that fall outside these limits are ignored. The remaining finite interval is divided up into equal partitions. The probability assigned to each interval is the observed relative frequency and the abscissa representing the interval is the observed conditional expectation. The problem with this method is that the discretized distribution does not converge towards the continuous distribution because of the truncation. Also, simulation is rather inefficient as accurate results can only be obtained for a very large number of simulations.

Let S_i be the number of realizations of the ith variable in the discretized distribution and let K be the number of simulations that were run for each distribution. Table 3.2 shows numerical results that were obtained by MATLAB using this method of simulation. Note that this solution method is a stochastic method, since two runs of the method with the same number of realizations and simulations will most probably *not* produce the same results. It can be seen that little benefit is derived from the very refined discretization. Even when each variable is approximated by 100 realizations that were obtained by $200,000$ simulations (so that in total the random vector has 10^8 realizations), the solution obtained is still quite far away from the true solution.

Theoretically-Based Discretization

If more accurate results are required, a more theoretically correct method of discretization should be used, in the sense that the discretized distribution should converge towards the continuous distribution. Suppose that we want to discretize the continuous distribution of the random variable $\tilde{\xi}$ into S points ξ_i with associated probabilities p_i, for $i = 1, \ldots, S$. Most methods of discretization use either intervals of equal length or intervals with equal probabilities. The following procedure (which uses class intervals of equal length) is just one of many discretization methods that could be used.

Divide the support into S intervals, with the $S+1$ endpoints

$$a_0 < a_1 < \ldots < a_S$$

which are chosen as follows:

- For finite support, $a < \xi < b$, choose $a_0 = a$, $a_S = b$, $a_i = a + \frac{i}{S}(b-a)$ for $i = 1, \ldots, S-1$.
- For non-negative support, $0 < \xi < \infty$, choose $a_0 = 0$, $a_{S-1} = F_{\tilde{\xi}}^{-1}\left(1 - \frac{1}{10S}\right)$, $a_S = \infty$, $a_i = \frac{i}{S-1} a_{S-1}$ for $i = 1, \ldots, S-2$.
- For infinite support, $-\infty < \xi < \infty$, choose $a_0 = -\infty$, $a_S = \infty$, $a_1 = F_{\tilde{\xi}}^{-1}\left(\frac{1}{10S}\right)$, $a_{S-1} = F_{\tilde{\xi}}^{-1}\left(1 - \frac{1}{10S}\right)$, $a_i = a_1 + \frac{i}{S-1}(a_{S-1} - a_1)$ for $i = 2, \ldots, S-2$.

The rationale behind the $\frac{1}{10S}$ is simply to ensure that the extreme points of the discretized distributions are sufficiently far into the tails of the distributions that they approximate, and to ensure that they move into the tails at a faster rate than if $\frac{1}{S}$ was used. The Sth interval is given by $a_{S-1} < \xi < a_S$. The discretized distribution is then given by the probabilities $p_S = \Pr\left[a_{S-1} < \tilde{\xi} < a_S\right]$ and the conditional expectations $\xi_S = E\left[\tilde{\xi} \mid a_{S-1} < \tilde{\xi} < a_S\right]$. See Appendix B.2 for these probabilities and conditional expectations for the Normal, exponential and uniform distributions. Table 3.3 shows numerical results obtained by MATLAB for increasingly refined discretizations, where each variable has S realizations, and the random vector has S^4 realizations.

Comparison of Discretization Methods

Figure 3-3 shows that the solutions obtained using the theoretically-based discretization appear to converge to the correct solution quite rapidly, while for the same number of realizations, the solutions obtained by the simulation method suggested by Kall & Wallace are not even close to the correct values. This shows that the solutions obtained when using a discrete approximation to the continuous distribution are highly dependent on the discretization process.

Table 3.3: Numerical Results for Theoretically-Based Discretization

S	S^4	x_1	x_2	z
5	625	39.648711	22.626918	153.153383
10	10,000	39.589350	22.347282	153.508986
15	50,625	39.435144	22.617298	153.663762
25	390,625	39.179875	22.873445	153.787340
50	6.25×10^6	39.152679	22.879152	153.836746
100	1.0×10^8	39.166394	22.855076	153.848200
200	1.6×10^9	39.173765	22.855958	153.852739
500	6.25×10^{10}	39.171546	22.857749	153.854457
∞[2]	∞	39.171293	22.857779	153.854750

Comparison with Numerical Integration

Figure 3-4 shows that the solutions obtained using the theoretically-based discretization are comparable with the solutions obtained by numerical integration. The solutions from the theoretically-based discretization which were plotted in Figure 3-3 are replicated in Figure 3-4, but on a scale that shows more detail. Both methods obtain solutions that seem to converge to the correct solution, as one would expect. However, the numerical integration method became slower and slower as more accuracy was required (as mentioned previously), while the theoretical discretization procedure was easy to apply and solve for many realizations, since each step simply involved the summation of more terms. The discretization method seems to be the best to apply in practice, particularly since for many variables, numerical integration becomes intractable, and many techniques are available for discretized distributions. The discretization method also requires far less effort than exact solution by analytical techniques.

[2]Exact solution

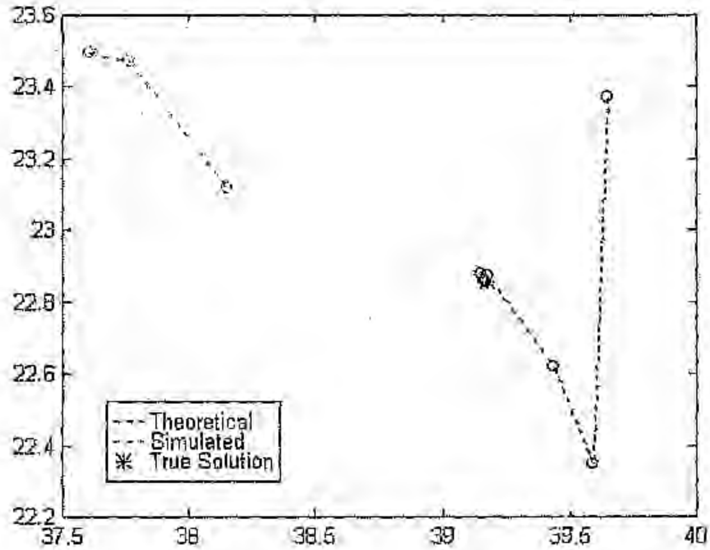

Figure 3-3: Comparison of Discretization Methods

3.7.8 Wait-and-See Solutions

The wait-and-see problem for a given ξ is

$$z(\xi) = \left\{ \begin{array}{ll} \min & 2x_1 + 3x_2 \\ \text{s.t.} & x_1 + x_2 \leq 100 \\ & (2 + \xi_3)x_1 + 6x_2 \geq 180 + \xi_1 \\ & 3x_1 + (3.4 - \xi_4)x_2 \geq 162 + \xi_2 \\ & x_1, x_2 \geq 0 \end{array} \right] \tag{3.55}$$

with the corresponding wait-and-see solution $\mathrm{x}(\xi) = (x_1(\xi), x_2(\xi))$. We want to find $\overline{z} = E\left[z(\tilde{\xi})\right] = WS$ and the average wait-and-see solution $\overline{\mathrm{x}} = (\overline{x}_1, \overline{x}_2) = E\left[\mathrm{x}(\tilde{\xi})\right]$. Note that there are no second-stage variables, and that there are some values of ξ for which the problem is infeasible (*e.g.* when at least one of ξ_1 and ξ_2 is extremely large), although the probability of

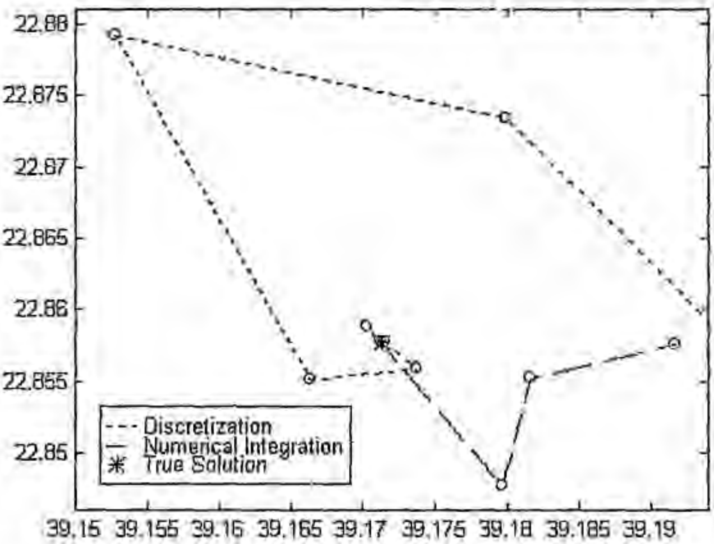

Figure 3-4: Comparison between Numerical Integration and Discretization

this occurring is minute. The expectations can be estimated by discretizing the distribution of $\tilde{\xi}$ or by simulation. First, we use the same theoretically-based discretization that was used in the previous section. If S denotes the number of realizations created for each random variable, the random vector will have S^4 different realizations. Table 3.4 shows the values of $\overline{z}$ and $\overline{x}$ that were calculated by MATLAB for different values of S, while the different wait-and-see solutions are plotted in Figure 3-5. Note that the wait-and-see solutions appear to be scattered in a systematic way, and convergence of $\overline{x}_1$, $\overline{x}_2$ and $\overline{z}$ appears to occur. Alternatively, we can simulate a number of independent realizations of the random vector. Let S denote the total number of simulations. Table 3.5 shows the values of $\overline{z}$ and $\overline{x}$ that were calculated by MATLAB for different numbers of simulations, and the wait-and-see solutions are plotted in Figure 3-6. Note that the wait-and-see solutions seem to be randomly scattered, while convergence of $\overline{x}_1$, $\overline{x}_2$ and $\overline{z}$ appears to be slower than when the theoretically-based discretization was used. Thus

Table 3.4: Average Wait-and-See Solutions using Theoretical Discretization

S	S^4	$\overline{x}_1$	$\overline{x}_2$	$\overline{z}$
5	625	36.283333	17.594259	125.349442
10	10,000	36.287524	17.583349	125.325096
20	160,000	36.287535	17.581078	125.318303
30	810,000	36.287288	17.580754	125.316836
40	2,560,000	36.287126	17.580670	125.316263

Table 3.5: Average Wait-and-See Solutions Calculated by Simulation

S	$\overline{x}_1$	$\overline{x}_2$	$\overline{z}$
100	36.425164	17.997990	126.844294
1,000	36.802746	17.213568	125.246196
10,000	36.341603	17.555148	125.348648
100,000	36.272115	17.596776	125.334559
1,000,000	36.281538	17.584998	125.318071

we estimate

$$WS = \overline{z} \simeq 125.316$$
$$\overline{x} \simeq (36.287\,,\, 17.581)$$

so that

$$EVPI = RP - WS \simeq 153.855 - 125.316 = 28.539$$

Recall that $VSS \simeq 85.210$, so that perfect information offers much less of an improvement over the stochastic solution than the stochastic solution offers over the deterministic solution. Now $AWS = z(36.287, 17.581) = 247.5933 > EEV \simeq 239.064$. Thus the average wait-and-see solution has a higher expected cost than the deterministic solution, even though it is far more difficult to compute. Although the average wait-and-see solution might seem intuitively like a good solution to use for the recourse problem, it has been shown to be a poor approach in terms of both computational effort and expected cost.

3.7.9 Model with Joint Chance Constraints

Suppose that the producer cannot simply buy the shortage of products in the market if the demand exceeds production, and thus when demand exceeds production, some demand will

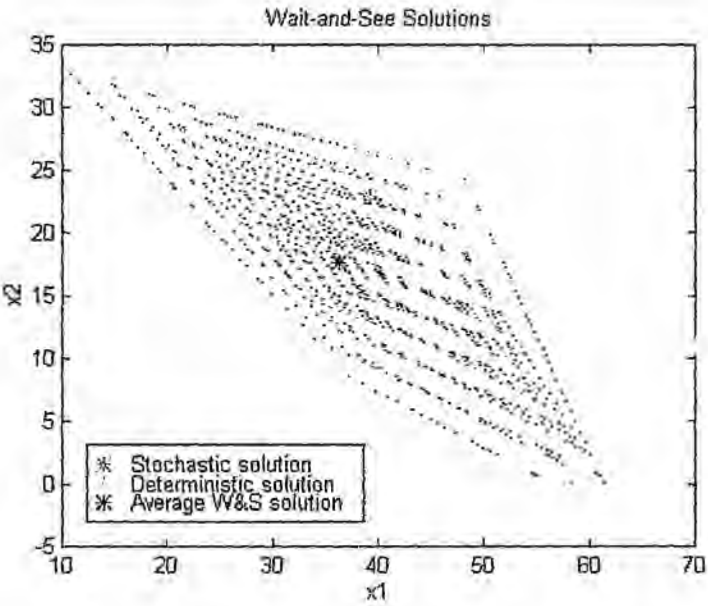

Figure 3-5: Wait-and-See Solutions for Theoretically-Based Discretization

have to go unmet. In order to maintain the client base, the producer may find it important to meet the clients' demands with a high reliability of, say, 95%. The producer is therefore faced with the chance-constrained problem

$$\left.\begin{array}{l} \min\ 2x_1 + 3x_2 \\ \text{s.t.}\ \ x_1 + x_2 \le 100 \\ \Pr\left[\begin{array}{c} (2+\tilde{\xi}_3)x_1 + 6x_2 \ge 180 + \tilde{\xi}_1 \\ 3x_1 + (3.4-\tilde{\xi}_4)x_2 \ge 162 + \tilde{\xi}_2 \end{array}\right] \ge 0.95 \\ \quad x_1 \ge 0\,,\ x_2 \ge 0 \end{array}\right\} \tag{3.56}$$

By independence of the $\tilde{\xi}_i$,

$$\Pr\left[(2+\tilde{\xi}_3)x_1 + 6x_2 \ge 180 + \tilde{\xi}_1 \text{ and } 3x_1 + (3.4-\tilde{\xi}_4)x_2 \ge 162 + \tilde{\xi}_2\right]$$

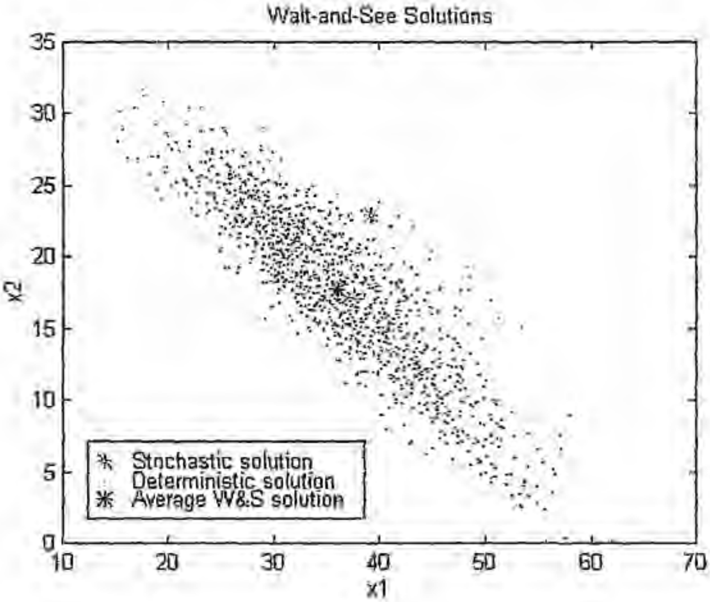

Figure 3-6: Wait-and-See Solutions Calculated by Simulation

$$
\begin{aligned}
&= \Pr\left[(2+\tilde{\xi}_3)x_1 + 6x_2 \geq 180 + \tilde{\xi}_1\right] \Pr\left[3x_1 + (3.4 - \tilde{\xi}_4)x_2 \geq 162 + \tilde{\xi}_2\right] \\
&= \Pr\left[\tilde{\xi}_1 - x_1\tilde{\xi}_3 \leq 2x_1 + 6x_2 - 180\right] \Pr\left[\tilde{\xi}_2 + x_2\tilde{\xi}_4 \leq 3x_1 + 3.4x_2 - 162\right]
\end{aligned}
$$

We obtain the distributions of $\tilde{Y}_1 := \tilde{\xi}_1 - x_1\tilde{\xi}_3$ and $\tilde{Y}_2 := \tilde{\xi}_2 + x_2\tilde{\xi}_4$ by transformation. For fixed x_1 and x_2, the density and distribution functions of $\tilde{Y}_1$ are

$$
f_1(y_1) = \frac{1}{1.6x_1}\left\{\Phi\left(\frac{y_1 + 0.8x_1}{12}\right) - \Phi\left(\frac{y_1 - 0.8x_1}{12}\right)\right\}
$$

and

$$
\begin{aligned}
F_1(y_1) &= \frac{7.5}{x_1}\left[\phi\left(\frac{y_1 + 0.8x_1}{12}\right) - \phi\left(\frac{y_1 - 0.8x_1}{12}\right)\right] \\
&\quad + \frac{7.5}{x_1}\left(\frac{y_1 + 0.8x_1}{12}\right)\Phi\left(\frac{y_1 + 0.8x_1}{12}\right) - \frac{7.5}{x_1}\left(\frac{y_1 - 0.8x_1}{12}\right)\Phi\left(\frac{y_1 - 0.8x_1}{12}\right)
\end{aligned}
$$

for $-\infty < y_1 < \infty$ and, for fixed x_1 and x_2 the density and distribution functions of $\widetilde{Y}_2$ are

$$f_2(y_2) = \frac{2.5}{x_2} \exp\left[\frac{253.125}{x_2^2} - \frac{2.5y_2}{x_2}\right] \Phi\left(\frac{x_2y_2 - 202.5}{9x_2}\right)$$

and

$$F_2(y_2) = \Phi\left(\frac{x_2y_2 - 202.5}{9x_2} + \frac{45}{2x_2}\right) - \exp\left[\frac{253.125}{x_2^2} - \frac{2.5y_2}{x_2}\right] \Phi\left(\frac{x_2y_2 - 202.5}{9x_2}\right)$$

for $-\infty < y_2 < \infty$. If we denote

$$F_1(y_1; x_1) := \Pr\left[\widetilde{\xi}_1 - x_1\widetilde{\xi}_3 \le y_1\right]$$

and

$$F_2(y_2; x_2) = \Pr\left[\widetilde{\xi}_2 + x_2\widetilde{\xi}_4 \le y_2\right]$$

then the reliability R of a given decision (x_1, x_2) can be written as

$$R(x_1, x_2) = F_1\left(2x_1 + 6x_2 - 180\,;\, x_1\right) F_2\left(3x_1 + 3.4x_2 - 162\,;\, x_2\right) \tag{3.57}$$

so that the problem with joint chance constraints becomes the nonlinear program

$$\left.\begin{array}{ll} \min & z = 2x_1 + 3x_2 \\ \text{s.t.} & x_1 + x_2 \le 100 \\ & F_1\left(2x_1 + 6x_2 - 180\,;\, x_1\right) F_2\left(3x_1 + 3.4x_2 - 162\,;\, x_2\right) \ge 0.95 \\ & x_1 \ge 0\,,\, x_2 \ge 0 \end{array}\right\} \tag{3.58}$$

The program (3.58) was solved by nonlinear programming methods using MATLAB to yield

$$\left.\begin{array}{l} x_1^* = 38.337403 \\ x_2^* = 24.538709 \\ z^* = 150.290934 \\ R(x_1^*, x_2^*) = 0.95 \end{array}\right\} \tag{3.59}$$

Note that the joint chance constraint is active at the solution. Figure 3-7 plots the feasible region and the solution of the above problem.

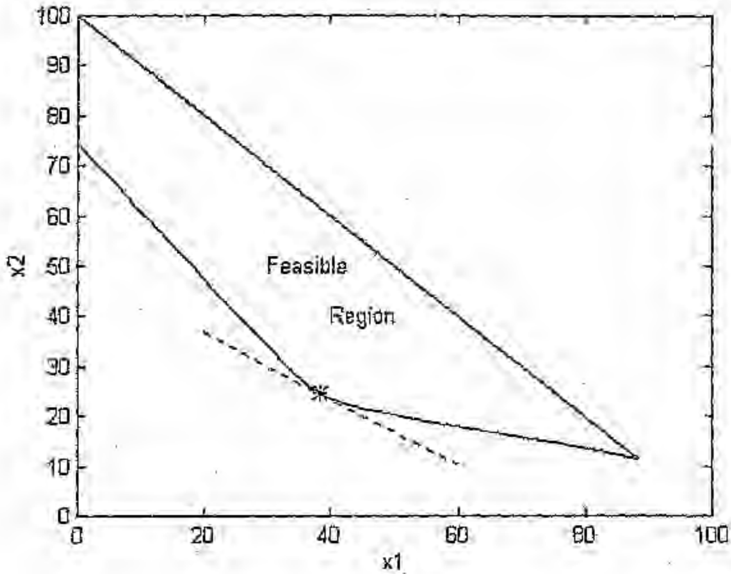

Figure 3-7: Feasible Region and Solution for Model with Joint Chance Constraints

3.7.10 Model with Separate Chance Constraints

It is quite likely that the producer will have different client bases for the different products. The clients of one product will not be concerned about the reliability level of the other product. The producer may thus require a high reliability level for each of the products, separately. The problem can then be formulated as a program with separate chance constraints.

$$
\left.
\begin{aligned}
\min \quad & z = 2x_1 + 3x_2 \\
\text{s.t.} \quad & x_1 + x_2 \leq 100 \\
& F_1\,(2x_1 + 6x_2 - 180\,;\, x_1) \geq 0.95 \\
& F_2\,(3x_1 + 3.4x_2 - 162\,;\, x_2) \geq 0.95 \\
& x_1 \geq 0\,,\, x_2 \geq 0
\end{aligned}
\right\} \tag{3.60}
$$

We define the reliability levels R_1 and R_2 of the two products as

$$
[illegible]\, x_2) \;=\; F_1\,(2x_1 + 6x_2 - 180\,;\, x_1) \tag{3.61}
$$

$$
[illegible]\, x_1 + 3.4x_2 - 162\,;\, x_2) \tag{3.62}
$$

The program (3.60) [illegible] ramming methods using MATLAB to obtain

$$
\left.
\begin{aligned}
& [illegible] \\
& x_2^* = 22.918456 \\
& z^* = 146.053047 \\
& R_1(x_1^*, x_2^*) = 0.95 \\
& R_2(x_1^*, x_2^*) = 0.95 \\
& R(x_1^*, x_2^*) = 0.9025
\end{aligned}
\right\} \tag{3.63}
$$

Note that both chance constraints are active at the optimal solution. The feasible region for the problem with separate chance constraints looks very similar to the feasible region for the problem with joint chance constraints, which was plotted in Figure 3-7.

3.7.11 Comparison of Solutions

We now have six different solutions: the expected value solution, the fat solution, the average wait-and-see solution, the recourse solution, and the solutions of the joint chance-constrained problem and the problem with separate chance constraints. We expect the solutions to the deterministic problems (*i.e.* the expected value solution and the fat solution) to be inferior when stochasticity is taken into account. We compare these solutions in Table 3.6 by evaluating

Table 3.6: Comparison of Solutions

Solution	x_1	x_2	$z(x_1,x_2)$	$R_1(x_1,x_2)$	$R_2(x_1,x_2)$	$R(x_1,x_2)$
Expected value solution	36	18	239.064	0.5	0.5259	0.2630
Fat solution	48.511	25.449	173.562	0.9998	0.9985	0.9982
Average W&S solution	36.287	17.581	247.593	0.4671	0.5102	0.2383
Recourse solution	39.171	22.858	153.855	0.9522	0.9572	0.9114
Joint CCP solution	38.337	24.538	155.159	0.9867	0.9628	0.95
Separate CCP solution	38.649	22.918	153.918	0.95	0.95	0.9025

their expected cost under the recourse problem and their reliability levels. The table shows that the solutions obtained by the stochastic models (*i.e.* the recourse model and the chance-constrained models) are good in terms of both expected cost and reliability. There is no solution that is optimal under the recourse model as well as the chance-constrained models, although the solutions are similar.

The recourse solution and the solutions to the chance-constrained problems (CCP solutions) are not similar purely by accident. In fact, if the minimum reliability levels of either of the chance-constrained models were chosen to be equal to the reliability levels resulting from the recourse solution, then the solution to that chance-constrained problem would be very close to the recourse solution. This is because the recourse solution would then be a point somewhere along the boundary of the feasible region of the chance-constrained problem, and the recourse objective (curved) and chance-constrained objective (linear) would be likely to produce similar optima. In this example, the recourse solution is closer to the separate CCP solution than the joint CCP solution since the recourse solution and separate CCP solution have reliability levels that are more similar.

The expected value solution and the average wait-and-see solution are very similar and are poor with respect to both expected cost and reliability. On the contrary, the fat solution leads to very high reliability and a reasonable expected cost which is due mainly to a negligible second-stage cost. The fat solution was calculated from a simple deterministic model and is thus quite a good solution with regard to the computational effort involved. It was not as bad as we expected and was actually better than the expected value solution.

3.8 Recourse Models *versus* Chance-Constrained Models

From a mathematical point of view, neither model is superior to the other. The more appropriate model is determined by the situation facing the decision maker and the preferences of the decision maker.

Recourse models are often favoured because more is known about them and their solution methods are well developed, while large chance-constrained models can be difficult to solve. Recourse models are justified when second-stage actions with their associated costs can be clearly defined and quantified. This is often possible. Recourse models bear no regard to how often the constraints are violated - they are only interested in the cost thereof.

Chance-constrained models on the other hand pr[illegible] no indication of the size of possible constraint violations and corresponding penalty costs. Nevertheless, there are many practical situations involving decision-making where reliability is regarded as the most important issue. This may be because it seems impossible to quantify a penalty, because of ethical reasons or to maintain an image. In such situations, a chance-constrained model may be the only appropriate model.

Chapter 4

Stochastic Nonlinear Programming

This chapter is concerned with the general form of a stochastic program - the stochastic nonlinear program. The chapter starts with the nonlinear formulation of stochastic programs with recourse and moves on to some basic properties of stochastic nonlinear programs. Solution methods and scenario modelling are then briefly discussed. The chapter ends with two worked examples of how stochastic nonlinear programs can be solvable and valuable.

4.1 Nonlinear Formulation

As stated in Section 1.5, the general *two-stage stochastic nonlinear program with recourse* can be written in the form

$$\min_{x \in X} \left\{ g_0(x) + E_{\tilde{\xi}} \left[Q\left(x, \tilde{\xi}\right) \right] \right\} \tag{4.1}$$

where the recourse function $Q(x, \xi)$ is defined as

$$Q(x, \xi) = \min_{y \in Y} \{ q(y) \mid h_i(y) \geq g_i(x, \xi) \ , \ h_i(y) \geq 0 \ , \ i = 1, \ldots, m \} \tag{4.2}$$

We call $x \in X \subseteq \mathbb{R}^n$ the decision vector and $g_0(x)$ the first-stage cost, where $g_0 : \mathbb{R}^n \to \mathbb{R}$. The recourse vector $y(\xi) \in Y \subseteq \mathbb{R}^{\overline{n}}$ consists of $\overline{n}$ recourse variables, and $q : \mathbb{R}^{\overline{n}} \to \mathbb{R}$ and $h_i : \mathbb{R}^{\overline{n}} \to \mathbb{R}$ are given functions. Note that (4.1) and (4.2) are identical to (1.9) and (1.8) which are repeated here for clarity. Once again, the recourse or second-stage decision y is typically different for each realization ξ, and it is therefore denoted $y(\xi)$. An alternative formulation is

$$\min_{x \in X} \{g_0(x) + \mathcal{Q}(x)\} \tag{4.3}$$

where $\mathcal{Q}(x)$ denotes the expected recourse function

$$\mathcal{Q}(x) = E_{\tilde{\xi}}\left[Q\left(x, \tilde{\xi}\right)\right] \tag{4.4}$$

The form (4.1) is known as *additive recourse*, which is an appropriate model provided that it is meaningful and acceptable to the decision maker to minimize total expected costs, *i.e.* the sum of first-stage and expected recourse costs. Other forms of the objective can be devised, such as objectives with multiplicative recourse. We only consider additive recourse since it is a realistic description in most situations.

A general model could include recourse as well as *probabilistic constraints*. Probabilistic constraints can be converted into equivalent deterministic constraints (which are usually nonlinear) by using the distribution functions of the constraints, as shown in Chapter 2. We assume that they are added to the constraint section of the recourse model so that we need only consider our general recourse model. Note, however, that the probabilistic constraints can easily define a nonconvex feasible region, as was discussed in Chapter 2.

4.2 Nonlinear Properties

4.2.1 Convexity

The following propositions show that stochastic programs with recourse define convex programs under fairly general conditions.

Proposition 17 *If $g_0(\cdot)$ is convex in x, $Q(\cdot, \xi)$ is convex in x $\forall \xi \in \Xi$, and X is a convex set, then the two-stage recourse program (4.1) is a convex program.*

Proof. For $x^1, x^2 \in X$ and $\overline{x} := \lambda x^1 + (1-\lambda) x^2$ where $\lambda \in (0,1)$ we have, by the convexity of $g_0(\cdot)$ and $Q(\cdot, \xi)$,

$$g_0(\overline{x}) + Q(\overline{x}, \xi) \leq \lambda \left[g_0(x^1) + Q(x^1, \xi)\right] + (1-\lambda)\left[g_0(x^2) + Q(x^2, \xi)\right] \ \forall \xi \in \Xi$$

which implies

$$g_0(\overline{x}) + E_{\tilde{\xi}}\left[Q\left(\overline{x}, \tilde{\xi}\right)\right] \leq \lambda \left(g_0(x^1) + E_{\tilde{\xi}}\left[Q\left(x^1, \tilde{\xi}\right)\right]\right) + (1-\lambda)\left(g_0(x^2) + E_{\tilde{\xi}}\left[Q\left(x^2, \tilde{\xi}\right)\right]\right)$$

and therefore (4.1) is a convex program since the objective is convex in x and X is a convex set. ■

Proposition 18 *If $g_0(\cdot)$ is convex in x, X is a convex set and in (4.2), $q(\cdot)$ is convex in y, $g_i(\cdot, \xi)$ is convex in x $\forall \xi \in \Xi$ for $i = 1, \ldots, m$, and $h_i(\cdot)$ is concave in y for $i = 1, \ldots, m$, then (4.1) is a convex program.*

Proof. Let $x^1, x^2 \in X$ and assume that y^1 and y^2 solve (4.2) for x^1 and x^2 respectively, at some realization ξ, so that $Q(x^1, \xi) = q(y^1)$ and $Q(x^2, \xi) = q(y^2)$. Then, by the convexity of g_i and the concavity of h_i for $i = 1, \ldots, m$, we have for any $\lambda \in (0,1)$,

$$\begin{aligned} g_i\left(\lambda x^1 + (1-\lambda)x^2, \xi\right) &\leq \lambda g_i\left(x^1, \xi\right) + (1-\lambda) g_i\left(x^2, \xi\right) \\ &\leq \lambda h_i(y^1) + (1-\lambda) h_i(y^2) \\ &\leq h_i\left(\lambda y^1 + (1-\lambda) y^2\right) \end{aligned}$$

Hence $\overline{y} := \lambda y^1 + (1-\lambda) y^2 \geq 0$ is feasible in (4.2) for $\overline{x} := \lambda x^1 + (1-\lambda) x^2$, and therefore we have, by the convexity of q,

$$\begin{aligned} Q(\overline{x}, \xi) &\leq q(\overline{y}) = q\left(\lambda y^1 + (1-\lambda) y^2\right) \\ &\leq \lambda q(y^1) + (1-\lambda) q(y^2) = \lambda Q\left(x^1, \xi\right) + (1-\lambda) Q\left(x^2, \xi\right) \end{aligned}$$

which shows that $Q(\cdot, \xi)$ is convex in x $\forall \xi$. Hence (4.1) is a convex program by the previous proposition, since $g_0(\cdot)$ is convex in x and X is a convex set. ■

4.2.2 Smoothness

For stochastic linear programs with recourse, discrete distributions induce a non-differentiable expected recourse function $\mathcal{Q}(\mathrm{x})$, and therefore we cannot expect differentiability of $\mathcal{Q}(\mathrm{x})$ in the general nonlinear case. However, the smoothness of recourse problems, or more precisely partial differentiability of the expected recourse function $\mathcal{Q}(\mathrm{x})$, can be asserted when $\widetilde{\xi}$ has a continuous distribution and certain conditions are satisfied.

Proposition 19 *If $Q(\mathrm{x},\widetilde{\xi})$ is partially differentiable with respect to x_j at some $\hat{\mathrm{x}}$ almost surely (i.e. for all ξ except possibly those belonging to an event $N_\delta \in \mathcal{F}$ with $P(N_\delta) = 0$), if its partial derivative $\frac{\partial Q(\hat{\mathrm{x}},\xi)}{\partial x_j}$ is integrable and if the residuum $\rho_j(\hat{\mathrm{x}}, \xi; h)$ satisfies*

$$\lim_{h\to 0} \frac{1}{h} \int_{\Xi} \rho_j(\hat{\mathrm{x}}, \xi; h)\, dP(\xi) = 0$$

then $\frac{\partial \mathcal{Q}(\hat{\mathrm{x}})}{\partial x_j}$ exists as well (i.e. $\mathcal{Q}(\mathrm{x})$ is partially differentiable at $\hat{\mathrm{x}}$ with respect to x_j) and

$$\frac{\partial \mathcal{Q}(\hat{\mathrm{x}})}{\partial x_j} = \int_{\Xi} \frac{\partial Q(\hat{\mathrm{x}}, \xi)}{\partial x_j}\, dP(\xi)$$

Proof. Let $\mathrm{e}_j = (0, \ldots, 0, \overbrace{1}^{j\text{th position}}, 0, \ldots, 0)^T$. By the definition of partial differentiability, the recourse function $Q(\mathrm{x},\xi)$ is partially differentiable with respect to x_j at $(\hat{\mathrm{x}}, \widehat{\xi})$ if there is a function $\frac{\partial Q(\mathrm{x},\xi)}{\partial x_j}$ such that

$$\frac{Q(\hat{\mathrm{x}} + h\mathrm{e}_j, \widehat{\xi}) - Q(\hat{\mathrm{x}}, \widehat{\xi})}{h} = \frac{\partial Q(\hat{\mathrm{x}}, \widehat{\xi})}{\partial x_j} + \frac{\rho_j\left(\hat{\mathrm{x}}, \widehat{\xi}; h\right)}{h}$$

with the residuum $\rho_j\left(\hat{\mathrm{x}}, \widehat{\xi}; h\right)$ satisfying

$$\lim_{h\to 0} \frac{\rho_j\left(\hat{\mathrm{x}}, \widehat{\xi}; h\right)}{h} = 0 .$$

Provided that $Q(\mathrm{x}, \widetilde{\xi})$ is partially differentiable at $\hat{\mathrm{x}}$ almost surely, we get

$$\frac{\mathcal{Q}(\hat{\mathrm{x}} + h\mathrm{e}_j) - \mathcal{Q}(\hat{\mathrm{x}})}{h} = \int_{\Xi} \frac{Q(\hat{\mathrm{x}} + h\mathrm{e}_j, \xi) - Q(\hat{\mathrm{x}}, \xi)}{h}\, dP(\xi)$$

$$= \int_{\Xi - N_\delta} \left[\frac{\partial Q(\hat{\mathrm{x}}, \xi)}{\partial x_j} + \frac{\rho_j(\hat{\mathrm{x}}, \xi; h)}{h} \right] dP(\xi)$$

$$= \int_{\Xi - N_\delta} \frac{\partial Q(\hat{\mathrm{x}}, \xi)}{\partial x_j} dP(\xi) + \int_{\Xi - N_\delta} \frac{\rho_j(\hat{\mathrm{x}}, \xi; h)}{h} dP(\xi)$$

where $N_\delta \in \mathcal{F}$ and $P(N_\delta) = 0$. nce the partial derivative of $\mathcal{Q}$ with respect to x_j is obtained by taking limits on both sides, *i.e.*

$$\frac{\partial \mathcal{Q}}{\partial x_j} = \lim_{h \to 0} \frac{\mathcal{Q}(\hat{\mathrm{x}} + h\mathrm{e}_j) - \mathcal{Q}(\hat{\mathrm{x}})}{h} = \int_{\Xi - N_\delta} \frac{\partial Q(\hat{\mathrm{x}}, \xi)}{\partial x_j} dP(\xi) + \lim_{h \to 0} \int_{\Xi - N_\delta} \frac{\rho_j(\hat{\mathrm{x}}, \xi; h)}{h} dP(\xi)$$

it follows that $\mathcal{Q}$ is partially differentiable at $\hat{\mathrm{x}}$ if $\int_{\Xi - N_\delta} \frac{\partial Q(\hat{\mathrm{x}}, \xi)}{\partial x_j} dP(\xi)$ exists (*i.e.* $\frac{\partial Q(\hat{\mathrm{x}}, \xi)}{\partial x_j}$ is integrable), and the residuum satisfies $\lim_{h \to 0} \frac{1}{h} \int_{\Xi - N_\delta} \rho_j(\hat{\mathrm{x}}, \xi; h)\, dP(\xi) = 0$. ∎

It is often possible to decide whether the recourse function is partially differentiable almost surely and whether the partial derivative is integrable. However, the requirement that the residuum be integrable and integral converges to zero faster than h can be difficult to check. Recalling that in the linear case, $\mathcal{Q}(\mathrm{x})$ is differentiable if the distribution is of continuous type (*i.e.* the cumulative distribution function is continuous), one may conjecture that nonlinear stochastic programs with continuous distributions are *likely* to be differentiable.

4.? Solution Methods

One of the simplest ways of solving a stochastic nonlinear program is to form the *deterministic equivalent* program and to solve it by nonlinear programming methods. This is often possible for programs where the distribution has a small number of realizations or scenarios. In some cases, it may be possible to work with the continuous distribution, if a closed form expression is available for $\mathcal{Q}(\mathrm{x})$ or if $\mathcal{Q}(\mathrm{x})$ is computable for a given x.

An efficient solution method exists for *stochastic quadratic programs* where both the first-stage and second-stage objective functions are quadratic. The method is a generalization of the L-shaped method. This can be compared to the way in which deterministic quadratic programs can be solved by a generalization of the simplex method. See Section 6.2 in Birge & Louveaux [10] for details of the algorithm.

Convex stochastic programs can be solved by a *stochastic quasi-gradient method*. The method generates a sequence of points $\{x^\nu\}$ that converges to an optimal solution of (4.1). The method uses search directions that depend on randomly generated samples. Stochastic quasi-gradient methods are stochastic methods and will produce a different sequence $\{x^\nu\}$ each time the algorithm is executed. See Section 3.9 in Kall & Wallace [28] or Section 10.3 in Birge & Louveaux [10] for details on stochastic quasi-gradient methods.

State-of-the-art methods for solving stochastic nonlinear programs based on Lagrangians aim to avoid the expensive evaluation of gradients of $Q(x)$, which can dominate a solution procedure. Such methods can solve large stochastic nonlinear programs and include the *progressive hedging algorithm* of Rockafellar & Wets [39] and the *basic Lagrangian dual ascent method*. Refer to Section 6.3 in Birge & Louveaux [10] for a development of these methods.

4.4 Scenario Modelling

The example in the next section shows that by constructing a fairly trivial statistical distribution using a small number of scenarios, the resulting stochastic model can offer an improvement over deterministic models. The major advantage of such *scenario modelling* is that the models can usually be solved without great difficulty, since the deterministic equivalent problem is not too large. Scenario modelling can often be useful when it is difficult to construct a meaningful statistical distribution objectively, or when a model that has many realizations (scenarios) is difficult to solve. Scenarios are often based on "expert" guesses or opinions that are *subjective* in nature.

4.5 Example in Kinematics

The problem in this section comes from an unsolved exercise at the end of Chap 1 in Birge & Louveaux [10]. It provides an illustrative example in motor racing kinematics, where scenario modelling with three scenarios is useful in creating a stochastic model that has value over the corresponding deterministic model. The value of the stochastic solution and the expected value of perfect information are calculated and it is shown that in this case, the stochastic solution is almost as valuable as having perfect information.

4.5.1 How Late can you Brake?

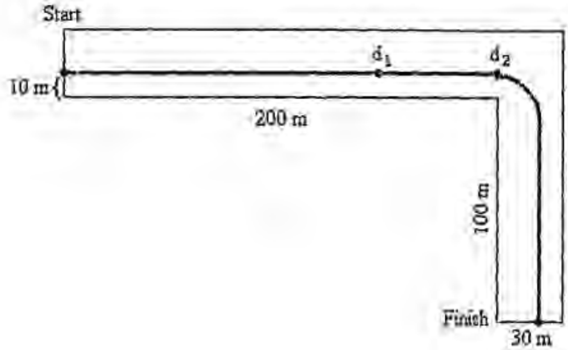

Figure 4-1: Opening Straight and Right-Hand Turn at "Suzuka"

After qualifying in pole position at a motor race at "Suzuka", you are trying to determine the quickest way to get through the first right-hand turn, which begins 200 metres from the start on a track that is 30 metres wide - see Figure 4-1. You aim to stay 10 metres inside the barrier on the opening straight and accelerate as fast as possible until a point d_1 metres from the end of the straight. At this point, you start to brake as hard as possible and take the turn in a circular arc at the current velocity that is reached at some point d_2 metres from the end of the straight. You take the turn with the tightest possible radius given your velocity and traction. A well-known equation in kinematics states that this radius is given by the square of the velocity divided by maximum lateral acceleration. Once you are through the turn you start once again to accelerate as fast as possible down the following straight. The section under consideration finishes 100 metres past the beginning of this straight.

The problem is that you cannot be sure of the combination of speed and traction that your car has on the track until you start braking at point d_1. For simplicity, we refer to this combination as track speed. At that point, you can tell whether the track is fast, medium or slow, and you can then determine the point d_2 where you enter the turn. You suppose that the three kinds of

Table 4.1: Parameters for Different Track Speeds

Random Variable	Fast ($s = 1$)	Medium ($s = 2$)	Slow ($s = 3$)	Expected Value
Acceleration ξ_1^s	$27\ m/s^2$	$24\ m/s^2$	$20\ m/s^2$	$23.\dot{6}\ m/s^2$
Deceleration ξ_2^s	$45\ m/s^2$	$42\ m/s^2$	$35\ m/s^2$	$40.\dot{6}\ m/s^2$
Max lateral acceleration ξ_3^s	$17.5\ m/s^2$	$16\ m/s^2$	$14\ m/s^2$	$15.8\dot{3}\ m/s^2$

track speeds are equally likely. You want to minimize your expected time through this section. You also hope that if you follow an optimal strategy, other competitors will not throw you out of the race. The track speeds correspond to scenarios that we denote by $s = 1, 2, 3$. The random variables that depend on the scenarios are acceleration $\widetilde{\xi}_1$, deceleration $\widetilde{\xi}_2$ and maximum lateral acceleration $\widetilde{\xi}_3$ (all measured in m/s^2) and hence only take on three possible values each, as given in Table 4.1.

4.5.2 Wait-and-See Model

To create a model, we divide the motion of the car into four sectors:

1. Acceleration from the start to d_1
2. Deceleration from d_1 to d_2
3. Circular turn through 90° at constant velocity
4. Acceleration after the turn to the end of the straight

Let t_i = time taken (in seconds) for the ith sector, $i = 1, 2, 3, 4$

v_i = instantaneous velocity (in metres/second) at point d_i, $i = 1, 2$

r = radius of turning arc

For the sectors with motion in a straight line, we use the well-known equations of motion

$$V = U + AT \tag{4.5}$$

$$V^2 = U^2 + 2AD \tag{4.6}$$

where U = initial velocity, V = final velocity, A = (constant) acceleration, D = displacement, and T = time taken. The parameters and constraints for these sectors are summarized in Table 4.2. Note that we define v_3 as the final velocity in the fourth sector.

Table 4.2: Parameters and Constraints for Sectors with Straight-Line Motion

	First Sector	Second Sector	Fourth Sector
Initial velocity U	0	v_1	v_2
Final velocity V	v_1	v_2	v_3
Acceleration A	ξ_1	$-\xi_2$	ξ_1
Displacement D	$200 - d_1$	$d_1 - d_2$	$110 - r$
Time taken T	t_1	t_2	t_4
Equation (4.5)	$v_1 = \xi_1 t_1$	$v_2 = v_1 - \xi_2 t_2$	$v_3 = v_2 + \xi_1 t_4$
Equation (4.6)	$v_1^2 = 2\xi_1(200 - d_1)$	$v_2^2 = v_1^2 - 2\xi_2(d_1 - d_2)$	$v_3^2 = v_2^2 + 2\xi_1(110 - r)$

The third sector must be treated differently. The sector time is t_3, the circular velocity is v_2 and as described above, the radius of the turning circle is $r = v_2^2/\xi_3$. Since the car turns through an arc of 90°, the distance travelled is $\frac{1}{2}\pi r$ which is equated with $v_2 t_3$, to get the constraint $v_2 t_3 = \frac{1}{2}\pi r$. We must also ensure that the entire turning arc lies on the track. It can be shown by (analytic) geometry that this condition is enforced by the constraint

$$d_2^2 + 100 - 20r \leq 0 \tag{4.7}$$

In addition, the constraint $0 \leq r - d_2 \leq 30$ ensures that the car ends up on the track for the next straight. In creating the model, further constraints are $0 \leq d_2 \leq d_1 \leq 200$ and non-negativity $t_i \geq 0$, $v_i \geq 0$, $r \geq 0$. The wait-and-see model for a given realization ξ is the nonlinear program

$$\left.\begin{array}{ll} \min & z = t_1 + t_2 + t_3 + t_4 \\ \text{s.t.} & v_1 = \xi_1 t_1 \,,\; v_1^2 = 2\xi_1(200 - d_1) \\ & v_2 = v_1 - \xi_2 t_2 \,,\; v_2^2 = v_1^2 - 2\xi_2(d_1 - d_2) \\ & v_2 t_3 = \frac{1}{2}\pi r \,,\; r = v_2^2/\xi_3 \\ & d_2^2 + 100 - 20r \leq 0 \\ & v_3 = v_2 + \xi_1 t_4 \,,\; v_3^2 = v_2^2 + 2\xi_1(110 - r) \\ & 0 \leq r - d_2 \leq 30 \,,\; 0 \leq d_2 \leq d_1 \leq 200 \\ & r, t_1, t_2, t_3, t_4, v_1, v_2, v_3 \geq 0 \end{array}\right\} \tag{4.8}$$

in the decision variables d_1 and d_2. For any given decision on d_1 and d_2, the other variables (*viz.* $r, t_1, t_2, t_3, t_4, v_1, v_2, v_3$) follow automatically as functions of d_1 and d_2. The expected value problem is equivalent to the wait-and-see problem (4.8) with the realization $\overline{\xi} = E(\tilde{\xi})$.

Table 4.3: Solutions to the Wait-and-See Problems

	Fast Track	Medium Track	Slow Track	Expected Value
z	7.836220	8.238503	8.914473	8.294105
d_1	88.72139	86.86101	86.47013	87.44365
d_2	34.49490	34.49490	34.49490	34.49490
r	64.49490	64.49490	64.49490	64.49490
t_1	2.871038	3.070545	3.369419	3.084121
t_2	0.976055	0.989753	1.066846	1.009060
t_3	3.015529	3.153716	3.371464	3.170272
t_4	0.973598	1.024489	1.106744	1.030651
v_1	77.58103	73.69309	67.38839	72.99087
v_2	33.59555	32.12348	30.04877	31.95575
v_3	59.88269	56.71123	52.18364	56.34782

The wait-and-see problems for each scenario and the expected value problem were solved using LINGO (see Appendix A.5) and the solutions are listed in Table 4.3.

Note that the widest possible turning radius (of $r = 64.49490$ metres) is chosen in each scenario. Another (suboptimal) local minimum to the expected value problem exists with $z = 8.428050$, $r = 9.356536$, $d_1 = 78.32423$, $d_2 = 9.33438$. This is a totally different line that involves braking later, turning much later with a very sharp radius, and finishing on the opposite side of the track. The difference in time is only $8.428050 - 8.294105 = 0.13395$ (*i.e.* 13 hundredths of a second), even though the lines are totally different. Figure 4-2 illustrates the optimal wait-and-see points d_1 and d_2 under each scenario for the optimal line, and also illustrates the aforementioned suboptimal line that the car can take. Note that the optimal line is the same under all scenarios, and that the braking points are so close together that it is difficult to distinguish between them when looking at the graph.

4.5.3 Recourse Formulation

We now develop the recourse problem. We must decide on d_1 in advance but we can observe ξ before deciding on d_2 and r. Although the decision d_1 must be taken before observation of ξ (*i.e.* before braking), the time taken t_1 to reach d_1 still depends on ξ, since the acceleration up

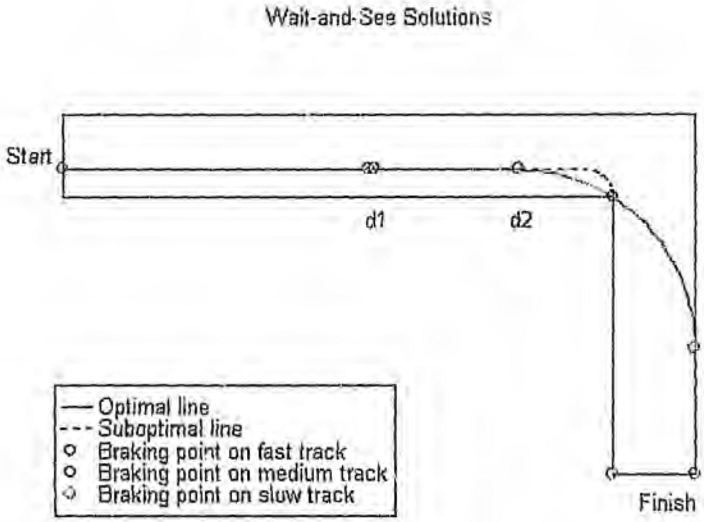

Figure 4-2: Optimal Braking Points under each Scenario

to that point depends on ξ, even though it cannot be observed. The recourse problem is

$$\left.\begin{array}{ll} \min & z = \frac{1}{3}\sum_{s=1}^{3}(t_1^s + t_2^s + t_3^s + t_4^s) \\ \text{s.t.} & v_1^s = \xi_1^s t_1^s\,,\ (v_1^s)^2 = 2\xi_1^s(200 - d_1)\,,\ \forall s \\ & v_2^s = v_1^s - \xi_2^s t_2^s\,,\ (v_2^s)^2 = (v_1^s)^2 - 2\xi_2^s(d_1 - d_2^s)\,,\ \forall s \\ & v_2^s t_3^s = \frac{1}{2}\pi r^s\,,\ r^s = (v_2^s)^2 / \xi_3^s\,,\ \forall s \\ & (d_2^s)^2 + 100 - 20r^s \leq 0\,,\ \forall s \\ & v_3^s = v_2^s + \xi_1^s t_4^s\,,\ (v_3^s)^2 = (v_2^s)^2 + 2\xi_1^s(110 - r^s)\,,\ \forall s \\ & 0 \leq r^s - d_2^s \leq 30\,,\ 0 \leq d_2^s \leq d_1 \leq 200\,,\ \forall s \\ & r^s,\ t_1^s,\ t_2^s,\ t_3^s,\ t_4^s,\ v_1^s,\ v_2^s,\ v_3^s \geq 0\,,\ \forall s \end{array}\right\} \qquad (4.9)$$

Table 4.4: Optimal Second-Stage Decisions for the Recourse Problem

	Fast ($s=1$)	Medium ($s=2$)	Slow ($s=3$)
d_2^s	25.79183	33.73613	34.49490
r^s	38.26093	63.73613	64.49490
t_1^s	2.899934	3.075845	3.369419
t_2^s	1.164939	0.997293	1.066846
t_3^s	2.322023	3.135110	3.371464
t_4^s	1.538123	1.041289	1.106744
T^s	7.925619	8.249537	8.914473
v_1^s	78.29823	73.82028	67.38839
v_2^s	25.87598	31.93396	30.04877
v_3^s	67.40531	56.92490	52.18364

with the optimal objective and first-stage solution obtained by LINGO being

$$\left.\begin{aligned} RP = z^* &= 8.363210 \\ d_1^* &= 86.47013 \end{aligned}\right\} \qquad (4.10)$$

Define the total time taken under scenario s as $T^s := \sum_{i=1}^{4} t_i^s$. The optimal second-stage solutions to (4.9) are given in Table 4.4. The optimal decision is to brake at the same point at which you would brake if you knew the track was slow. On a slow track, the decision is therefore optimal. On medium and fast tracks, you brake for a longer time (and distance) and take the corner with a tighter radius than you would if you knew the track speed. The optimal braking point and lines to take under the recourse problem are illustrated in Figure 4-3. Note that the lines for the medium and slow tracks are very close together and in fact correspond for part of the final straight, so that in the graph the blue line is hidden by the green line.

4.5.4 The Value of the Scenario Model

The expected value of perfect information can be calculated as

$$\begin{aligned} WS &= \frac{1}{3}(7.836220 + 8.238503 + 8.914473) = 8.329732 \\ EVPI &= RP - WS = 8.363210 - 8.329732 = 0.033478 \end{aligned}$$

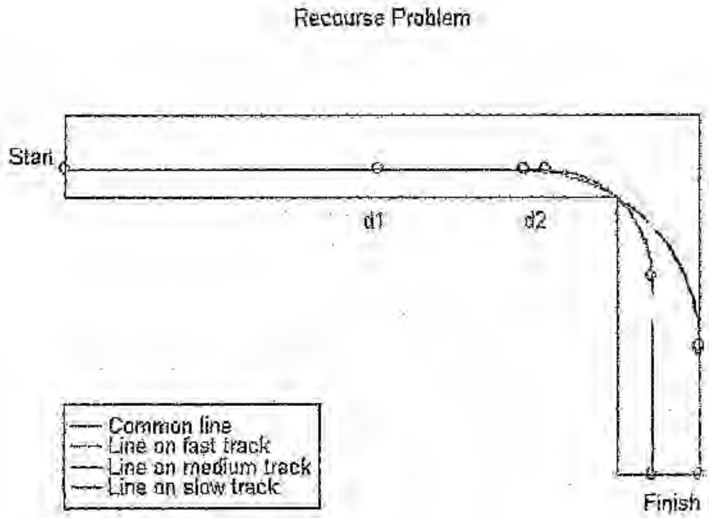

Figure 4-3: Optimal Braking Point and Lines under the Recourse Problem

so that perfect information is expected to improve the time through the section (relative to the recourse solution) by 33 thousandths of a second, which is a reduction in time of $0.033478 / 8.363210 = 0.4\%$.

In order to evaluate the expected result of using the expected value solution, fix the first-stage decision $d_1 = 87.44365$ and choose the second-stage decisions optimally to calculate $EEV = 8.492229$. The value of the stochastic solution is therefore

$$VSS = EEV - RP = 8.492229 - 8.363210 = 0.129019$$

corresponding to a reduction in time through the section (relative to the expected value solution) of 129 thousandths of a second, or $0.129019 / 8.492229 = 1.52\%$. Note that in this example, $VSS \simeq 4EVPI$ and therefore using the stochastic solution is almost as good as having perfect information. Table 4.5 shows the total time through the section under each

Table 4.5: Total Times under each Scenario for each Solution

	Fast Track	Medium Track	Slow Track	Average Time
Expected Value Solution	7.870619	8.402066	9.204002	$8.429229 = EEV$
Recourse Solution	7.925619	8.249537	8.914473	$8.363210 = RP$
Difference	−0.055000	0.152529	0.289529	$0.129019 = VSS$
Recourse Solution	7.925619	8.249537	8.914473	$8.363210 = RP$
Wait-and-See Solution	7.836220	8.238503	8.914473	$8.329732 = WS$
Difference	0.089399	0.011034	0.000000	$0.033478 = EVPI$

scenario for the wait-and-see solutions, the recourse solution and the expected value solution. The recourse solution is slightly worse than the expected value solution when the track is fast but is significantly better for slow and medium tracks. When compared with the wait-and-see solutions, the recourse solution is optimal for a slow track, slightly suboptimal for a medium track, and rather more suboptimal for a fast track, yet it is fairly good under all three scenarios. Note that in practice, however, we cannot take the wait-and-see solutions, but must make one decision here and now. This example clearly shows that the recourse solution is robust across all scenarios - it is the best solution that we can take on average given the information that we have.

The average wait-and-see solution for d_1 is $\overline{d}_1 = \frac{1}{3}(88.72139 + 86.86101 + 86.47013) = 87.35084$ and has an expected total time of $AWS = 8.474030$. Note that in this example, $RP < AWS < EEV$, so that the average wait-and-see solution is better than the expected value solution (but, of course, worse than the recourse solution).

4.6 Example in Manufacturing Design

This section provides a worked example of a nonlinear stochastic program with nonlinear recourse and a continuous distribution that depends on the decision taken. The example was published in [7] and subsequently in Section 1.4 of Birge & Louveaux [10] as an example to illustrate the value of a stochastic program. However, I found that the published value of EEV was incorrect[1], and this led to a gross underestimation of VSS, the value of the stochastic solution. The example has been used with permission from Prof. John R. Bir of the University

[1] See http://www-personal.umich.edu/~jrbirge/book.html for errata to [10].

of Michigan, U.S.A.

4.6.1 Design of an Axle

Consider the design of a go-cart axle. The designer must determine the length w and diameter ξ of the axle, both of which are measured in inches. Together, these quantities determine the performance characteristics of the product. The goal is to determine a combination that gives the gr... est expected profit.

However, when the axle is produced, the actual dimensions are not exactly those that were specified. We suppose that the length w can be produced exactly, but that the diameter ξ is a random variable $\tilde{\xi}(x)$ that depends on the setting x on a machine that is used in the production process. Note that this differs from our usual assumption in stochastic programming that the random vector $\tilde{\xi}$ is independent of the decision taken x.

4.6.2 Recourse Model

We assume a symmetric triangular distribution for $\tilde{\xi}(x)$ on $[0.9x, 1.1x]$. This distribution has the density function

$$f(\xi; x) = \left\{ \begin{array}{cl} \frac{100}{x^2}(\xi - 0.9x) & \text{if } 0.9x \leq \xi \leq x \\ \frac{100}{x^2}(1.1x - \xi) & \text{if } x \leq \xi \leq 1.1x \\ 0 & \text{otherwise} \end{array} \right] \tag{4.11}$$

The decisions w and x are subject to certain limits, $w \leq w_{\max}$ and $x \leq x_{\max}$. We assume that all the axles produced can be sold. From marketing studies, it has been found that the selling price depends on the length w and can be expressed as

$$r\left(1 - e^{-0.1w}\right) \tag{4.12}$$

where r is the selling price of a very long axle. An axle of length w and diameter ξ has volume $\pi\left(\frac{\xi}{2}\right)^2 w$. We assume, however, that the cost of producing the axles is determined by the setting x on the machine (rather than by ξ) because material is acquired before the actual machining process. The manufacturing cost is therefore

$$\frac{cw\pi x^2}{4} \tag{4.13}$$

where c is the cost per unit volume of the material.

Other costs, known as *quality losses*, are incurred after the product is made due to warranty claims and potential future sales losses from pr luct defects. In our model, these costs are the recourse costs. The go-cart will perform poorly if the axle becomes bent or broken due to excess stress or deflection. From well-known results in physics, it follows that the stress limit for an axle that is made of steel with a breaking stress of $10,000\, lbs/in^2$ and a maximum central load of 100 lbs, is

$$\frac{w}{\xi^3} \leq 39.27 \tag{4.14}$$

For deflection, we use a maximum speed of 2000 $r.p.m.$ (corresponding to a speed of roughly 60 km/h for a typical 15 cm wheel) and a maximum deflection of $0.1\, in$. Once again, it can be shown from well-known results in physics that this leads to the equation

$$\frac{w^3}{\xi^4} \leq 63,169 \tag{4.15}$$

When either of these constraints is violated, the axle deforms. The cost for not meeting these constraints is assumed proportional to the square of the violation. We express it as

$$Q(w, x, \xi) = \min_{y \geq 0} \left\{ qy^2 \mid \frac{w}{\xi^3} - y \leq 39.27,\ \frac{w^3}{\xi^4} - 300y \leq 63,169 \right\} \tag{4.16}$$

where y is the maximum of stress violation and (to maintain similar units) $\frac{1}{300}$ of the deflection violation. This is an unusual way to define the recourse function. A more obvious way to define the recourse function would be to define one recourse variable y_1 corresponding to the stress violation and another recourse variable y_2 corresponding to the deflection violation. The recourse function would then be described as a function of these two recourse variables. The expected recourse cost, given w and x, is

$$\begin{aligned}\mathcal{Q}(w,x) &= \int_{-\infty}^{\infty} Q(w,x,\xi)f(\xi;x)d\xi \qquad (4.17)\\ &= q\int_{0.9x}^{x}\left(\frac{100}{x^2}\right)(\xi-0.9x)\left[\max\left\{0,\frac{w}{\xi^3}-39.27,\frac{w^3}{300\xi^4}-210.56\dot{3}\right\}\right]^2 d\xi\\ &+q\int_{x}^{1.1x}\left(\frac{100}{x^2}\right)(1.1x-\xi)\left[\max\left\{0,\frac{w}{\xi^3}-39.27,\frac{w^3}{300\xi^4}-210.56\dot{3}\right\}\right]^2 d\xi\end{aligned}$$

Our aim is to maximize expected profit per item, which equals total revenue per item less manufacturing cost per item less expected future cost per item. Mathematically, we obtain the nonlinear deterministic equivalent program

$$\begin{aligned}\max z(w,x) &= r\left(1-e^{-0.1w}\right)-c\left(\frac{w\pi x^2}{4}\right)-\mathcal{Q}(w,x)\\ \text{s.t. } 0 &\le w\le w_{\max},\ 0\le x\le x_{\max} \qquad (4.18)\end{aligned}$$

For our problem, let $w_{\max}=36$, $x_{\max}=1.25$, $r=10$, $c=0.025$ and $q=1$. The program (4.18) can be solved by standard methods of nonlinear programming, provided that we can evaluate $\mathcal{Q}(w,x)$. The calculation of $\mathcal{Q}(w,x)$ would be greatly simplified if the expression involving the maximum of three quantities could be expressed as the maximum of two quantities. In order to do this, we assume that the deflection constraint (4.15) is satisfied in the neighbourhood of the optimal solution. This implies that $\frac{w^3}{300\xi^4}-210.56\dot{3}\le 0$ for $0.9x\le\xi\le 1.1x$ for all (w,x) in the neighbourhood of the optimal solution (w^*,x^*), and hence

$$\max\left\{0,\frac{w}{\xi^3}-39.27,\frac{w^3}{300\xi^4}-210.56\dot{3}\right\}=\left\{\begin{array}{cc}\frac{w}{\xi^3}-39.27 & \text{if } \xi<\xi' \\ 0 & \text{if } \xi\ge\xi'\end{array}\right] \qquad (4.19)$$

where $\xi'=\sqrt[3]{\frac{w}{39.27}}$. We then solve the recourse program under assumption (4.19) and check that the assumption is satisfied. It is shown later that this assumption is indeed satisfied in the neighbourhood of the stochastic solution, but not in the neighbourhood of the deterministic solution - this might be why Birge [7] erred in his calculation of EEV. In order to obtain a simple closed form expression for $\mathcal{Q}(w,x)$, we must consider the two cases $0.9x\le\xi'\le x$ and

$x < \ \le 1.1x$ separately. We first consider the case

$$0.9x \le \xi' \le x \tag{4.20}$$

Then

$$\begin{aligned}\mathfrak{Q}(w,x) &= \int_{0.9x}^{\xi'} \left(\frac{100}{x^2}\right)(\xi - 0.9x)\left(\frac{w}{\xi^3} - 39.27\right)^2 d\xi + \int_{\xi'}^{1.1x} 0\, d\xi \\ &= 30032.885\frac{w^{2/3}}{x^2} - 7.6207895\frac{w^2}{x^6} - 4363.\dot{3}\frac{w}{x^3} - 73499.297\frac{w^{1/3}}{x} + 62456.382\end{aligned}$$

The solution to (4.18) was obtained by LINGO as

$$\left.\begin{aligned} z^* &= 8.935328 \\ w^* &= 33.46394 \\ x^* &= 1.037378 \end{aligned}\right\} \tag{4.21}$$

Now $\frac{w^{*3}}{300\xi^4} - 210.56\dot{3}$ is a decreasing function of ξ and is therefore maximized by $\xi^* = 0.9x^*$, when $\frac{w^{*3}}{300\xi^{*4}} - 210.56\dot{3} = -46.163793 < 0$. By perturbing w^* and x^* we can see that the deflection constraint (4.15) is indeed satisfied for all ξ for (w,x) in the neighbourhood of the optimal solution (4.21), *i.e.* assumption (4.19) is satisfied. Furthermore, assumption (4.20) is satisfied since $\xi' = 0.948066$ and $0.9x^* = 0.933640 < \xi' < 1.037378 = x^*$. Since both of these assumptions are satisfied, (4.21) is at least a local minimum of (4.18) and therefore we do not bother to consider the case $x < \xi' \le 1.1x$, since it is unlikely that this case will yield an optimal solution.

4.6.3 Deterministic Relaxation

Consider the expected value problem where the random variable ξ is replaced by its mean x to obtain the deterministic problem

$$\begin{aligned}\max\ z(w,x,\bar{\xi}) &= r\left(1 - e^{-0.1w}\right) - c\left(\frac{w\pi x^2}{4}\right) - \left[\max\left\{0, \frac{w}{x^3} - 39.27, \frac{w^3}{300x^4} - 210.56\dot{3}\right\}\right]^2 \\ \text{s.t. } 0 &\le w \le w_{\max},\ 0 \le x \le x_{\max}\end{aligned} \tag{4.22}$$

Note that (4.22) allows for violations of the stress and deflection constraints. Any violations of these constraints are penalized accordingly. Models that allow for *constraint violations* are therefore not unique to stochastic programming. However, (4.22) ignores the randomness inherent in the problem. It is the *inclusion of randomness* in the model that distinguishes stochastic models from deterministic models.

Problem (4.22) was initially solved using LINGO, but the solution differed slightly from the solution obtained from MATLAB. By experimenting with different tolerances in the MATLAB solver, I noticed it was difficult to find an accurate solution to (4.22), since the problem seems to be badly scaled - this may be caused by the way in which the recourse function (4.16) was defined. We can, however, be sure of the following digits, obtained by MATHEMATICA (see Appendix A.5) and confirmed by MATLAB.

$$\left.\begin{aligned} \overline{z} &= 9.0615995 \\ \overline{w} &= 34.9702 \\ \overline{x} &= 0.962039 \end{aligned}\right\} \tag{4.23}$$

It may appear at first glance that this solution obtains a better expected profit than the stochastic solution, since $\overline{z} > z^*$. Once again, this is simply because the deterministic problem paints an overly optimistic picture of the actual situation.

4.6.4 The Value of the Stochastic Model

The main difficulty in finding EEV is in evaluating $Q(\overline{w}, \overline{x})$. The value of EEV was calculated incorrectly as 5.88 in [7] and [10]. Assumption (4.19) does not hold (*i.e.* the deflection constraint does not always hold) in the neighbourhood of the deterministic solution and therefore the maximum of the three quantities must be considered as follows.

$$\frac{\overline{w}}{\xi^3} - 39.27 > 0 \Rightarrow \xi < 0.9620827$$

and

$$\frac{\overline{w}^3}{300\xi^4} - 210.56\dot{3} > 0 \Rightarrow \xi < 0.9070842$$

Also,

$$\frac{\overline{w}}{\xi^3} - 39.27 > \frac{\overline{w}^3}{300\xi^4} - 210.56\dot{3} \Rightarrow 51388\xi^4 + 300\overline{w}\xi - \overline{w}^3 > 0 \Rightarrow \xi > 0.8975455$$

We are interested in the range $0.9\overline{x} < \xi < 1.1\overline{x}$, *i.e.* $0.8658351 < \xi < 1.0582429$ and therefore

$$\max\left\{0, \frac{\overline{w}}{\xi^3} - 39.27, \frac{\overline{w}^3}{300\xi^4} - 210.56\dot{3}\right\} = \left\{\begin{array}{ll} \frac{\overline{w}^3}{300\xi^4} - 210.56\dot{3} & \text{if } 0.8658351 < \xi \leq 0.8975455 \\ \frac{\overline{w}}{\xi^3} - 39.27 & \text{if } 0.8975455 < \xi \leq 0.9620827 \\ 0 & \text{if } 0.9620827 < \xi < 1.0582429 \end{array}\right]$$

We can now calculate $\mathcal{Q}(\overline{w}, \overline{x})$ as

$$\begin{aligned} \mathcal{Q}(\overline{w}, \overline{x}) &= \frac{100}{\overline{x}^2} \int_{0.8658351}^{0.8975455} (\xi - 0.9\overline{x}) \left(\frac{\overline{w}^3}{300\xi^4} - 210.56\dot{3}\right)^2 d\xi \\ &\quad + \frac{100}{\overline{x}^2} \int_{0.8975455}^{0.962039} (\xi - 0.9\overline{x}) \left(\frac{\overline{w}}{\xi^3} - 39.27\right)^2 d\xi \\ &\quad + \frac{100}{\overline{x}^2} \int_{0.962039}^{0.9620827} (1.1\overline{x} - \xi) \left(\frac{\overline{w}}{\xi^3} - 39.27\right)^2 d\xi \\ &= 33.364116 \end{aligned}$$

and hence

$$z(\overline{w}, \overline{x}) = r\left(1 - e^{-0.1\overline{w}}\right) - c\left(\frac{\overline{w}\pi\overline{x}^2}{4}\right) - \mathcal{Q}(\overline{w}, \overline{x}) = -24.302488 = EEV$$

which is vastly different from the published value of 5.88. The value of the stochastic solution is

$$VSS = RP - EEV = z^* - z(\overline{w}, \overline{x}) = 8.935328 - (-24.302488) = 33.237816$$

In this case, the stochastic formulation causes an improvement from a large *loss* under the deterministic solution to an actual profit. In the sense of proportionate improvement, the stochastic solution has infinite value relative to the deterministic solution. The expected value of perfect information (EVPI) was not calculated for this example.

Chapter 5

Stochastic Integer Programming

This chapter provides a brief overview of stochastic integer programming. It starts with the standard formulation of stochastic integer programs and their properties. Some of the difficulties associated with stochastic integer programming are discussed, and so are solution methods. A simple example illustrates that stochastic integer programs can be solvable and valuable. The example also illustrates how two-stage recourse can be used to model a repetitive long term problem, and a discussion of this concludes the chapter. See Appendix A.4 for background in deterministic integer programming.

5.1 Integer Formulation of the Recourse Problem

Stochastic integer programs with recourse are stochastic recourse programs where some or all of the decision variables are *integers*. When referring to stochastic integer programs, it is normally understood that we are referring to stochastic integer *linear* programs, although of course stochastic nonlinear programs can have integer decision variables too. The standard form of the stochastic integer program

$$\min_{\mathbf{x}\in X} \left\{ \mathbf{c}^T\mathbf{x} + E_{\tilde{\xi}} \left[Q(\mathbf{x}, \tilde{\xi}) \right] \right\} \tag{5.1}$$

where

$$Q(\mathbf{x}, \xi) = \min_{\mathbf{y}\in Y} \left\{ \mathbf{q}(\xi)^T\mathbf{y} \mid W\mathbf{y} = \mathbf{h}(\xi) - T(\xi)\mathbf{x} \right\} \tag{5.2}$$

is the same as the standard form of the stochastic linear program (3.3) and (3.4) except that the sets X and Y are now defined as

$$\begin{aligned} X &:= \{\mathbf{x} \mid A\mathbf{x} = \mathbf{b}\,,\, \mathbf{x} \geq 0,\, x_i \in \mathbb{Z}\; \forall i \in I_1\} \\ Y &:= \{\mathbf{y} \mid \mathbf{y} \geq 0\,,\, y_i \in \mathbb{Z}\; \forall i \in I_2\} \end{aligned} \tag{5.3}$$

where I_1 is the set of subscripts of first-stage variables with integer restrictions and I_2 is the set of subscripts of second-stage variables with integer restrictions. The set Y can also contain further linear constraints. Thus in the formulation of stochastic integer programs, the sets X and Y can place *integrality* restrictions and *binary* restrictions on the variables $\mathbf{x}$ and $\mathbf{y}$, in addition to linear constraints. The definitions of $\mathbf{c}$, $\mathbf{b}$, A, $\widetilde{\boldsymbol{\xi}}$, W, T, and $\mathbf{h}$ are exactly as in Section 3.2. The program can also be written as the deterministic equivalent problem

$$\min_{\mathbf{x} \in X} \{\mathbf{c}^T\mathbf{x} + \mathcal{Q}(\mathbf{x})\} \tag{5.4}$$

by representing the second-stage problem implicitly as the *expected recourse function*

$$\mathcal{Q}(\mathbf{x}) = E_{\widetilde{\xi}}\,[Q(\mathbf{x}, \xi)]\ . \tag{5.5}$$

When the sets X and Y are defined as

$$\begin{aligned} X &= \{\mathbf{x} \mid A\mathbf{x} = \mathbf{b}\,.\, \mathbf{x} \in \mathbb{Z}_+^{n_1}\} \\ Y &= \mathbb{Z}_+^{n_2} \end{aligned} \tag{5.6}$$

where $\mathbb{Z}_+$ denotes the set of non-negative integers, we have a *stochastic pure integer program*. Of course, it is not necessary for all the decision variables to be integers. We could have a mixture of continuous and integer variables - such a situation is known as a *stochastic mixed integer program*.

5.2 Properties and Difficulties

Exactly as for linear recourse programs, we define the second-stage feasibility set or induced feasibility set $K = \{\mathbf{x} \mid \mathcal{Q}(\mathbf{x}) < \infty\}$, and the second-stage feasibility set $K(\xi) = \{\mathbf{x} \mid Q(\mathbf{x}, \xi) < \infty\}$

for a given realization ξ. Now if integrality restrictions feature in X, but there are none in Y, the properties of $\mathcal{Q}(x)$ and K are clearly the same as in the continuous case. However, when integrality constraints are present in the second-stage problem, the properties of $\mathcal{Q}(x)$ and K change and become very undesirable. The following proposition shows that such stochastic integer programs do not, in general, have the desirable properties of convexity and continuity. The proposition is illustrated in Section 3.3 of Birge & Louveaux [10] by means of examples.

Proposition 20 *The recourse function $Q(x, \xi)$ and hence the expected recourse function $\mathcal{Q}(x)$ of a stochastic integer program with second-stage integrality restrictions is, in general, nonconvex and discontinuous. Furthermore, the second-stage feasibility sets $K(\xi)$ and hence K are, in general, nonconvex sets.*

Since K is the intersection of a number of generally nonconvex sets $K(\xi)$, it is also, in general, nonconvex. Discontinuity of $Q(x, \xi)$ occurs in jumps corresponding to changes in the second-stage variables from one integer to the next. Discontinuity of $\mathcal{Q}(x)$ follows for finite distributions since $\mathcal{Q}(x)$ is a weighted average of a finite number of discontinuous functions. The discontinuity of these functions immediately implies their nonconvexity. However, it can be shown (see Stougie [42]) that when the random variable is of continuous type, the expected recourse function $\mathcal{Q}(x)$ is continuous, yet it remains nonconvex in general.

5.3 Solution Methods for Integer Problems

It is often possible to solve the deterministic equivalents of small stochastic integer programs using standard methods of integer programming. Integer programs start to become computationally intractable when there are more than about 100 integer variables. Problems can therefore arise in solving the deterministic equivalent when there are integer second-stage variables, because a different integer variable is required for each second-stage decision.

The sets X and Y are immediately nonconvex if they contain integrality constraints. Thus, with reference to the previous section, it can be seen that stochastic integer programs do not possess ideal properties (such as convexity or even continuity) for designing an efficient algorithmic procedure. Nevertheless, algorithms to solve stochastic integer programs can be devised by combining principles from integer programming (such as branch-and-bound and cutting-plane techniques) with methods for solving stochastic linear programs with continuous decision variables. Such an algorithm is an extension of the L-shaped method, known as the *integer L-shaped method*, and solves stochastic integer programs with first-stage integer variables. At each stage of the integer L-shaped algorithm, the *current problem*

$$
\begin{aligned}
\min \ z &= c^T x + \theta \\
\text{s.t. } Ax &= b \\
\gamma_\ell x &\geq \delta_\ell, \ \ell = 1, \ldots, s \qquad (5.7)\\
\beta_\ell x + \theta &\geq \alpha_\ell \quad \ell = 1, \ldots, t \qquad (5.8)\\
x &\geq 0 \,, \ \theta \in \mathbb{R}
\end{aligned}
$$

is solved. The current problem is obtained by:

- Relaxing the integer constraints $x^{(1)} \in \mathbb{Z}_+^{n_1}$ on the first stage variables.
- Relaxing the induced constraints $x \in K$ on the first stage variables and representing these constraints by *feasibility cuts* of the form (5.7).
- Relaxing the exact representation of $\mathcal{Q}(x)$ by an approximate polyhedral representation involving θ using *optimality cuts* of the form (5.8).

The integrality constraints are ultimately enforced by *branching and bounding* in the algorithm. When integrality is not satisfied, two problems are created by branching, or special feasibility cuts called *integrality cuts* are added when *branching and cutting*. At each iteration, the current problem becomes a more accurate representation of the exact problem (over a smaller subset of the feasible region), until either the node is fathomed or further branching occurs. By relaxing the integrality constraints, a linear program (approximately representing the stochastic program) is solved at each node. In the algorithm, it is assumed that for fixed x, $\mathcal{Q}(x)$ is computable in a finite number of steps.

Integer L-Shaped Algorithm

Step 0 Set $s = t = \nu = 0$, $\bar{z} = \infty$. The value of θ is set to $-\infty$ or to an appropriate lower bound and is ignored in the computation until optimality cuts are added. A list is created that initially contains only a single pendant node corresponding to the initial subproblem.

Step 1 Select some pendant node in the list as the current problem; if none exists, stop.

Step 2 Set $\nu = \nu + 1$. Solve the current problem. If the current problem has no feasible solution, fathom the current node as infeasible and go to Step 1. Otherwise, let $(\mathbf{x}^\nu, \theta^\nu)$ be an optimal solution.

Step 3 Check for any relaxed constraint violation. If one exists, add one feasibility cut (5.7), set $s = s + 1$, and go to Step 2. If $\mathbf{c}^T\mathbf{x}^\nu + \theta^\nu > \bar{z}$, fathom the current problem as bounded and go to Step 1.

Step 4 Check for integrality restrictions. If a restriction is violated, create two new nodes by a branch-and-bound procedure. Append the new nodes to the list of pendant nodes, and go to Step 1.

Step 5 Compute $\mathcal{Q}(\mathbf{x}^\nu)$ and $z^\nu = \mathbf{c}^T\mathbf{x}^\nu + \mathcal{Q}(\mathbf{x}^\nu)$. If $z^\nu < \bar{z}$, update $\bar{z} = z^\nu$.

Step 6 If $\theta^\nu \geq \mathcal{Q}(\mathbf{x}^\nu)$, then fathom the current node and return to Step 1. Otherwise, impose one optimality cut (5.8), set $t = t + 1$, and return to Step 2.

The optimality cuts and feasibility cuts are the same as those in the standard L-shaped method. A multicut approach is often preferred. When there are integer restrictions in the second stage, feasibility and optimality cuts based on second-stage branch-and-bound can be added when the random variable is discrete. See Section 8.1 in Birge & Louveaux [10] for a full development of the integer L-shaped method.

5.4 Airline Planning Example

The example in this section comes from an unsolved exercise at the end of Chapter 1 of Birge & Louveaux [10]. This example illustrates how a stochastic integer program with a few scenarios can be modelled and solved and that it is more valuable than the corresponding deterministic model. It also illustrates a situation where a two-stage stochastic program provides a solution to a long-term problem, as is explained in Section 5.5.

5.4.1 Partitioning an Aeroplane into Seating Classes

An airline is trying to decide how to partition a new plane for its new route. The plane can seat 200 economy class passengers. A section can be partitioned off for first class seats but each of these seats takes the space of two economy class seats. A business class section can also be included, but each of these seats takes as much space as $1\frac{1}{2}$ economy class seats. The profit on a first class ticket is, however, three times the profit of an economy ticket. A business class ticket has a profit of two times an economy ticket's profit. Once the plane is partitioned into these seating classes, it cannot be changed and it therefore corresponds to a first-stage decision. Let the first-stage decision variables be

x_1 = number of first class seats

x_2 = number of business class seats

x_3 = number of economy class seats

The airline knows that the plane will not always be full in each section. It has predicted that three scenarios (denoted by s) will occur with about the same frequency, and hence we assume each scenario to be equally likely:

1. Weekday morning and evening traffic ($s = 1$)
2. Weekend traffic ($s = 2$)
3. Weekday midday traffic ($s = 3$)

Let ξ_1^s = demand for first class seats under scenario s

ξ_2^s = demand for business class seats under scenario s

ξ_3^s = demand for economy class seats under scenario s

Table 5.1 gives the demand for tickets for each seating class under each scenario, as well as the mean demand for each seating class.

5.4.2 Two-Stage Integer Recourse Model

The airline wants to maximize its profits from this plane over a long period of time, such as a year. The probabilities of the scenarios correspond to the long-term relative frequencies of the scenarios. The second-stage decisions variables are

Table 5.1: Seating Demand under each Scenario

Seating Class	$s=1$	$s=2$	$s=3$	Mean
First Class ξ_1^s	20	10	5	$11.\dot{6}$
Business Class ξ_2^s	50	25	10	$28.\dot{3}$
Economy Class ξ_3^s	200	175	150	175

Table 5.2: Optimal Decisions to the Recourse Problem

Seating Class	x_i^*	y_i^{1*}	y_i^{2*}	y_i^{3*}
First Class ($i=1$)	10	10	10	5
Business Class ($i=2$)	20	20	20	10
Economy Class ($i=3$)	150	150	150	150

$y_1^s =$ no. of first class tickets sold under scenario s

$y_2^s =$ no. of business class tickets sold under scenario s

$y_3^s =$ no. of economy class tickets sold under scenario s

The airline does not allow overbooking, so that the number of tickets sold is limited by the number of seats available in each section. The recourse model can be written as

$$\left.\begin{aligned} \min \quad & z = -\tfrac{1}{3}\sum_{s=1}^{3}(3y_1^s + 2y_2^s + y_3^s) \\ \text{s.t.} \quad & 2x_1 + 1.5x_2 + x_3 \leq 200 \\ & y_i^s \leq x_i \,,\, y_i^s \leq \xi_i^s \,,\ \forall i, s \\ & x_i \in \mathbb{Z}_+ \ \forall i \,,\ y_i^s \in \mathbb{Z}_+ \ \forall i, s \end{aligned}\right\} \tag{5.9}$$

with the optimal objective $z^* = -208.\dot{3}$. The optimal decisions for the recourse problem were obtained using LINGO (see Appendix A.5) and are given in Table 5.2. Note that in solving (5.9), the second-stage integrality constraints $y_i^s \in \mathbb{Z}_+$ can be ignored since the optimal second-stage decisions are automatically integral. This is because the solver chooses $y_i^{s*} = \max(x_i, \xi_i^s)$, which is non-negative and integral since $x_i, \xi_i^s \in \mathbb{Z}_+$.

5.4.3 Deterministic Model and Solution

The expected value problem is

$$\left.\begin{array}{ll} \min & z=-3y_1-2y_2-y_3 \\ \text{s.t.} & 2x_1+1.5x_2+x_3\le 200 \\ & y_i\le x_i,\, y_i\le \bar{\xi}_i,\ \forall i \\ & x_i,\, y_i\in \mathbb{Z}_+\ \forall i \end{array}\right\} \tag{5.10}$$

and the optimal solution to the expected value problem by calculated using LINGO as

$$\left.\begin{array}{l} z=EV=-225 \\ x_1=y_1=11 \\ x_2=y_2=28 \\ x_3=y_3=136 \end{array}\right\}. \tag{5.11}$$

In solving this problem, the second-stage integrality constraints cannot be ignored, since not all $\bar{\xi}_i\in\mathbb{Z}_+$.

5.4.4 Value of the Stochastic Model

We fix the first-stage decisions as the optimal solutions to the expected value problem $x_1=11$, $x_2=28$, $x_3=136$ in (5.9) to get $EEV=-204$. The value of the stochastic solution is then

$$VSS=EEV-RP=-204-(-208.\dot{3})=4.\dot{3}$$

corresponding to an increase of $4.\dot{3}\,/\,204=2.1\%$ over the profit under the deterministic solution. Of course, this figure relies on the assumption that the three scenarios explain the variation in demand reasonably accurately.

5.5 Two-Stage Recourse in a Different Context

There is a subtle difference between the above example and the previous examples in this text which were modelled by two-stage recourse problems. In the above example, the randomness does not correspond to a single observation of a scenario (as in the other examples), but to a series of observations of scenarios. Each of the scenarios occurs with a relative frequency that does not change over time. It may even be known before each observation exactly which scenario will occur. The first-stage decision can be taken with regard to the scenarios that will occur, but it is a decision that must be reused many times. In the above example, the same seating arrangement must be used for every flight and cannot be changed between flights even though we know what the demand is going to be. The two-stage recourse model is entirely valid in this situation.

However, the concept of perfect information as we know it is no longer meaningful. This is because even if we knew exactly when each scenario would occur, we would still have to take the same first-stage decision every time. In other words, even though we have perfect information, we cannot take perfect decisions (*i.e.* the wait-and-see solutions). We must take one decision here and now, and once it has been made, it cannot be changed in response to the scenario that occurs. The optimal decision we can take is therefore the decision that is the best in the long run (*i.e.* the best on average), and this corresponds to the recourse solution. Thus "perfect information" in this sense effectively enables us to take a recourse decision.

Chapter 6

A Real-Life Application: The Unit Commitment Problem

This chapter provides a real-life application of stochastic programming in unit commitment. Eskom - one of the worl' s largest electricity producers - is faced with the unit commitment problem on a daily basis. A slightly simplified version of Eskom's problem for a period of one day is modelled first deterministically and then by a stochastic program. The models are solved and the value of the stocha model over both the deterministic model and Eskom's current strategy is determined. I- to acknowledge Dr John Dean [17] of Eskom (National Control) for providing data and information.

6.1 Background in Unit Commitment

Suppliers of electricity aim to mee, the continuous demand for electricity by generating a sufficient supply of electricity using their generating units. These units may be thermal, such as coal-fired and nuclear power stations, or hydro-electric. Reserve generating units, such as gas turbines or pumped storage, can also be used. If demand exceeds supply, a low system frequency situation will arise and eventually demand will be unmet. This is undesirable due to possible damage to electrical equipment, loss of potential income, safety and security considerations, opportunity cost to the consumer and loss of goodwill. The unit commitment problem is the problem of scheduling which of the generating units should be operated and their respective output levels during a certain time period. The generating units should meet the demand for electricity at each point in time, at the lowest total production cost, without violating their technical constraints [12].

At any time, some plants are on-line or committed (operating), while others are off-line (*i.e.* not operating - possibly undergoing maintenance or not required in meeting the demand). Plants that are currently on-line may fail randomly or be taken out of service on a planned basis. It is advisable that some units run below their maximum capacities so that the loading can be increased if other units fail or the demand increases unexpectedly. The resulting excess system capacity is called *spinning reserve*. The thermal units are subject to operating constraints such as minimum on-time (time spent on-line), minimum off-time (time spent off-line) and maximum ramping (*i.e.* the maximum rate at which the plant's output can be increased up to its maximum). Each of the thermal units must operate above its minimum capacity, below which the unit cannot run unless it is off.

Consider meeting the demand for electricity over the next day. The demand for electricity should be satisfied during each hour. We assume that demand is constant over each hour - this is a standard assumption that is used in unit commitment models [12]. Each unit (power station) has a number of sets (independent generators) that are identical. The decisions to switch sets on or off are taken every 24 hours. For a period of 24 hours following these decisions, each set will either remain off or remain on (assuming that it doesn't fail), and will operate at some level between minimum and maximum capacity in the latter case.

According to the recent paper of Takriti, Birge & Long [45], previous models used for solving the unit commitment problem were deterministic. These models assumed that the demand for any time period was known in advance, and this demand was obtained by forecasting. However, such forecasts are subject to statistical error and hence it may be valuable to create a stochastic model that allows for variable demand. The stochastic model in this chapter is based on [45].

6.2 The Problem Facing Eskom

Eskom is South Africa's national electricity utility and the world's fourth largest by capacity, generating over half of the electricity in Africa. The majority of Eskom's generating units are coal-fired power stations. Eskom also possesses a nuclear power station, hydro-electric units, gas turbines, a pumped storage scheme and interruptible load contracts.

We are interested in Eskom's unit commitment problem. In this study we work with time units of an hour, and we ignore ramping since Eskom's generating units can generally increase from minimum to maximum loading within an hour. Operating costs for thermal units include fuel costs and start-up costs. The fuel consumption of thermal units differs at various production levels. The shut-down costs of Eskom's thermal units are negligible.

In practice, the majority of the demand is met by coal-fired power stations. In fact, Eskom's generating capacity is so large that the coal-fired power stations alone could usually meet the demand without any of the other units being used. The model in this chapter includes all the thermal units, *i.e.* the coal-fired power stations and the Koeberg nuclear power plant. Gas turbines are used as reserve generating units. The hydro-electric plants and pumped-storage schemes are not included, as I was not given data on these units. It must be pointed out, however, that the generating capacity of these units is small compared to the capacities of the thermal units, and thus the exclusion of these units does not lead to much simplification of the actual problem under consideration. Interruptible load contracts are held with some industries that utilize large amounts of electricity. Such contracts allow Eskom to cut the supply to these industries at certain times (usually peak times), thus enabling it to meet the (now reduced) demand that it could not otherwise meet. This is known as demand-side management as opposed to supply-side management. Interruptible load is also excluded from the model, as it is only available for about two hours per week at most.

6.3 Model Parameters

Let i index the units, $i = 1, \ldots, 12$.

n_i = number of sets in unit i

t index the hours, $t = 1, \ldots, 24$

u_i' = number of sets in unit i that are on initially

g_i = minimum operating level (in MW) for each set in unit i

G_i = maximum operating level (in MW) for each set in unit i

Table 6.1: Unit Constraint Data

Unit No. i	Name of Unit	Unit Type	Number of Sets n_i	Initial State u_i'	Minimum Load g_i	Maximum Load G_i	Start-up Cost h_i
1	Arnot	coal	3	2	200	330	35000
2	Duvha	coal	6	4	350	575	59600
3	Hendrina	coal	10	7	130	190	17500
4	Kendal	coal	6	4	320	640	32100
5	Kriel	coal	6	4	255	475	31400
6	Lethabo	coal	6	4	375	593	47000
7	Matimba	coal	6	6	360	615	60500
8	Matla	coal	6	6	325	575	23100
9	Tutuka	coal	6	6	275	585	25700
10	Majuba	coal	2	0	306	612	44000
11	Koeberg	nuclear	2	2	920	920	0[1]
12	Gas Turbines	reserve	6	0	0	57	0[2]

h_i = start-up cost (in R) of each set in unit i

Table 6.1 contains data on the constraints under which each unit must operate.

The cost functions are calculated from the following data.

Let α_i = average heat rate (in GJ/MWh) at maximum loading for each set in unit i

γ_i = incremental heat rate (in GJ/MWh) for each set in unit i

β_i = no-load heat (in GJ/h) for each set in unit i

ϕ_i = fuel cost (in R/GJh^{-1}) for each set in unit i

As is shown in Figure 6-1, by equating two formulae for the heat rate at maximum loading, i.e. $G_i\alpha_i = \beta_i + G_i\gamma_i$, we obtain β_i as

$$\beta_i = (\alpha_i - \gamma_i)G_i \tag{6.1}$$

The cost of operating a set of unit i at level x MW for one hour is then $(\beta_i + \gamma_i x)\phi_i = \beta_i\phi_i + \gamma_i\phi_i x$, which is an affine function in x. For a unit that has u_i sets on, the cost of running that *unit* at the *total* output level x_i (i.e. x_i is the total over all sets of unit i) for an hour will

[1] Not applicable since it never shuts down.

[2] Reserve units have negligible start-up costs.

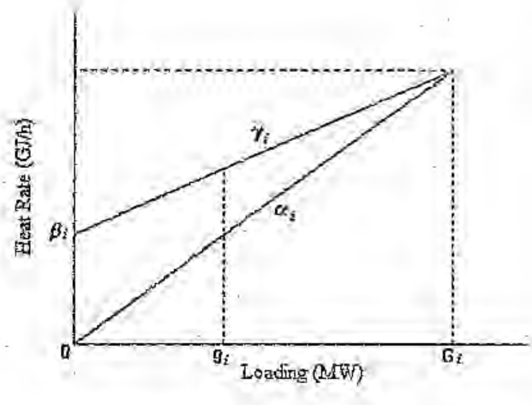

Figure 6-1: Calculation of No-Load Heat

be the linear function in u_i and x_i

$$\beta_i\phi_i u_i + \gamma_i\phi_i x_i = \rho_i u_i + \tau_i x_i \tag{6.2}$$

where $\rho_i = \beta_i\phi_i$ = fixed cost (in R/h) per set of unit i

and $\tau_i = \gamma_i\phi_i$ = marginal cost (in R/MWh) for unit i

Data on the operations of each unit is given in Table 6.2.

6.4 Deterministic Formulation

Under the deterministic model or expected value problem, we assume that the demand for each hour of the day is known accurately in advance. This is not a realistic assumption but it simplifies the modelling process. Eskom has a method of forecasting the demand and in the expected value problem we assume that these forecasts are perfectly accurate. Table 6.3 shows the demand forecasts that were obtained by Eskom for each hour of Monday, 15 January 1996, where hour 1 represents the hour between midnight and 1 a.m. *etc.*

Let $\overline{d}_t$ = demand forecast for hour t

u_i = number of sets in unit i that are on during the next 24-hour period

Table 6.2: Unit Operational Data

Unit No. i	Name of Unit	Average Heat Rate α_i	Incremental Heat Rate γ_i	No-Load Heat β_i	Fuel Cost ϕ_i	Fixed Cost ρ_i	Marginal Cost τ_i
1	Arnot	10.57	9.51	349.80	0.733	270.40	7.35
2	Duvha	10.33	9.46	500.25	0.535	267.63	5.06
3	Hendrina	11.14	10.23	172.90	0.655	113.25	6.70
4	Kendal	10.36	9.42	601.60	0.788	474.06	7.42
5	Kriel	10.22	9.17	498.75	0.801	399.50	7.35
6	Lethabo	10.06	9.56	296.50	0.981	290.87	9.38
7	Matimba	10.29	9.26	633.45	0.284	179.90	2.63
8	Matla	10.14	9.32	471.50	0.902	425.29	8.41
9	Tutuka	9.79	8.76	602.55	0.761	458.54	6.67
10	Majuba	10.87	10.04	507.96	2.710	1376.57	27.21
11	Koeberg	10.00	10.00	0	1.400	0	14.00
12	Gas Turbines	12.50	12.50	0	39.40	0	492.50

Table 6.3: Demand Forecasts

Hour t	Demand (MW) $\bar{d}_t$	Hour t	Demand (MW) $\bar{d}_t$
1	15527	13	20967
2	15386	14	20711
3	15250	15	20582
4	15348	16	20786
5	15653	17	20595
6	16882	18	20113
7	18832	19	20205
8	20424	20	21096
9	21365	21	21281
10	21365	22	20155
11	21093	23	18756
12	21157	24	17708

x_{it} = output level (in MW) at which unit i operates during hour t (total over all sets)

The decision variables for the expected value problem are the u_i and the x_{it}. All the parameters of the model, including demand, are known at the start of the day and therefore the entire schedule can be worked out at the start of the day. It is important to understand that since the sets within each unit are identical, we can take the decisions u_i and x_{it} at the *unit level*. Once these unit-level decisions have been taken, the decisions at the *set level* within the units can be scheduled automatically. Each $u_i \in \{0, \ldots, n_i\}$ is an integer decision variable that is limited by the number of sets available within the unit.

Each x_{it} is a non-negative decision variable. For any given $u_i > 0$, $g_i u_i \le x_{it} \le G_i u_i \ \forall t$. If $u_i = 0$, then this also holds since the condition $x_{it} = 0$ is enforced.

$$\therefore \ g_i u_i \le x_{it} \le G_i u_i \ \forall i, t \ \text{ and } \ x_{it} \ge 0 \ \forall i, t \tag{6.3}$$

The cost of switching on the sets for unit i is

$$\begin{cases} \left(u_i - u_i'\right) h_i \,, & \text{if } u_i - u_i' \ge 0 \\ 0 \,, & \text{if } u_i - u_i' < 0 \end{cases} \tag{6.4}$$

We define the variable $v_i = \max(u_i - u_i', 0)$ for $i = 1, \ldots, 12$ so that the start-up cost of all the units for the day is $\sum_{i=1}^{12} h_i v_i = \sum_{i=1}^{10} h_i v_i$ since $h_{11} = h_{12} = 0$. The constraints

$$v_i \ge u_i - u_i' \text{ and } v_i \ge 0 \tag{6.5}$$

together imply that $v_i = \max(u_i - u_i', 0)$, since the only expression involving v_i that appears in the objective function is $\sum h_i v_i$ and the objective function is minimized.

The cost of running all the units for the 24-hour period is

$$\sum_{i=1}^{12} \sum_{t=1}^{24} (\rho_i u_i + \tau_i x_{it}) = 24 \sum_{i=1}^{10} \rho_i u_i + \sum_{i=1}^{12} \sum_{t=1}^{24} \tau_i x_{it} \tag{6.6}$$

since $\rho_{11} = \rho_{12} = 0$.

We formally include unmet demand as the 13th unit, indexed by $i = 13$. It has a zero start-up cost, *i.e.* $p_{13} = 0$ and a high, penalizing marginal cost of $r_{13} = 10000$, say. We also define the parameters $n_{13} = 1$, $g_{13} = 0$ and $G_{13} = \infty$; and the decision variables

$x_{13,t}$ = amount of unmet demand during hour t, $0 \le x_{13,t} < \infty$

u_{13} = binary variable taking on the value 1 if there is any unmet demand and 0 otherwise.

There are some special constraints that must be taken into account. Both sets at Koeberg must run and at least three sets at Tutuka must run. We define ℓ_i as the minimum number of sets that must run at unit i, so that $u_i \in \{\ell_i, \ldots, n_i\}$. Therefore $\ell_9 = 3$, $\ell_{11} = 2$ and $\ell_i = 0\ \forall i \notin \{9, 11\}$.

The objective is to minimize the sum of the cost of starting up additional sets, the fixed costs of running the sets, and the incremental production costs. The constraints are that demand must be met[3] and the operating levels are restricted by minimum and maximum levels, depending on the u_i. The expected value problem may thus be written as the following mixed-integer linear program:

$$\left.\begin{array}{ll} \min & z = \sum_{i=1}^{10} h_i v_i + 24 \sum_{i=1}^{10} p_i u_i + \sum_{i=1}^{13} \sum_{t=1}^{24} r_i x_{it} \\ \text{s.t.} & v_i \ge u_i - u_i^* \ , \ v_i \ge 0 \ , \ i = 1, \ldots, 10 \\ & g_i u_i \le x_{it} \le G_i u_i \ , \ \forall i, t \\ & \sum_{i=1}^{13} x_{it} \ge \bar{d}_t \ , \ \forall t \\ & x_{it} \ge 0 \ , \ \forall i, t \\ & u_i \in \{\ell_i, \ldots, n_i\} \ , \ \forall i \end{array}\right\} \quad (6.7)$$

The model has $10 + 13 + 13 \times 24 = 335$ decision variables (of which 13 are integral and a further 10 automatically integral) and $10 + 2 \times 13 \times 24 + 24 = 658$ constraints.

6.5 Stochastic Formulation

Under the stochastic model or recourse problem, we assume that the demand vector is not known in advance and is a random vector characterized by a discrete statistical distribution, represented by a number of scenarios.

Let p_s = probability of occurrence of scenario s

[3] of which some may be allocated as unmet

Table 6.4: Scenario Probabilities

z	$\phi(z)$	s	p_s
-2	0.053991	1	0.054489
-1	0.241971	2	0.244201
0	0.398942	3	0.402620
1	0.241971	4	0.244201
2	0.053991	5	0.054489
Σ	0.990866	Σ	1.000000

d_t^s = demand (in MWh) during hour t under scenario s

u_i = number of sets in unit i that are on during the next 24-hour period

x_{it}^s = output level (in MW) at which unit i operates during hour t under scenario s

Note that u_i is a first-stage variable that does not depend on the scenario s. For the purpose of simplicity, we decided to represent the 24-hour demand vector by a 5-point distribution. We assume that the errors in the demand forecasts are multiplicative and perfectly autocorrelated over the 24-hour period - this is not a totally realistic assumption but was chosen for simplicity. It is estimated that the standard deviation of the error in each forecast is 2.5% of the mean of the forecast. Therefore, actual demand for every hour is, as illustrated in Figure 6-2:

- two standard deviations lower than the forecast, under scenario 1, *i.e.* $d_t^1 = 0.95\overline{d}_t\ \forall t$
- one standard deviation lower than the forecast, under scenario 2, *i.e.* $d_t^2 = 0.975\overline{d}_t\ \forall t$
- equal to the forecast, under scenario 3, *i.e.* $d_t^3 = \overline{d}_t\ \forall t$
- one standard deviation higher than the forecast, under scenario 4, *i.e.* $d_t^4 = 1.025\overline{d}_t\ \forall t$
- two standard deviations higher than the forecast, under scenario 5, *i.e.* $d_t^5 = 1.05\overline{d}_t\ \forall t$

We obtain the scenario probabilities from the density function of the standard Normal distribution by scaling up the ordinates $\phi(z)$ of the density at $z = -2, -1, 0, 1, 2$ so that they sum to 1, as in Table 6.4. The rationale for using this discrete 5-point distribution was so that we could obtain probabilities for the values $0.95\overline{d}_t$, $0.975\overline{d}_t$, $\overline{d}_t$, $1.025\overline{d}_t$, $1.05\overline{d}_t$. There is no other theoretical foundation behind the distribution. It must be emphasized that this distribution was chosen for simplicity and does not quite represent the true error structure accurately.

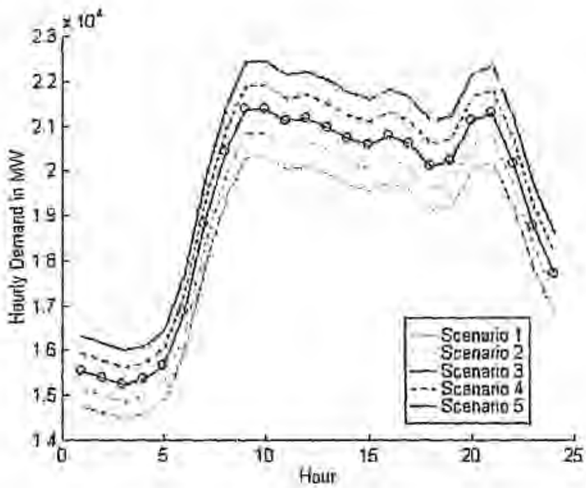

Figure 6-2: Demand under each Scenario

The decision variables for the recourse problem are the u_i and the x^s_{it}. Once again, since the sets within each unit are identical, we can take the decisions u_i and x^s_{it} at the unit level. Each u_i is a decision that must be made before observation of the random vector (which is the series of hourly demands) and hence u_i is a first-stage, integer decision variable that is independent of demand. The hourly decisions to be made for each unit for each hour depend on the observed demand, and therefore each x^s_{it} is a second-stage, non-negative decision variable that is dependent on the demand. In this model, d^s_t is known perfectly for all 24 hours at the end of the first hour, after which the problem is deterministic, and merely involves finding the optimal schedule for the sets that are already on-line. In this way, a 24-hour problem has been collapsed into a two-stage problem.

The objective is to minimize the sum of the cost of starting up additional sets, the fixed costs of running the sets, and the expected value of the incremental production costs. The constraints are that demand must be met[4] under all possible scenarios and the operating levels are restricted by minimum and maximum levels, depending on the u_i's. The recourse problem may thus be

[4] of which some may be allocated as unmet

written as the following mixed-integer linear program:

$$\left.\begin{array}{ll}\min & z=\sum_{i=1}^{10} h_i v_i + 24\sum_{i=1}^{10} p_i u_i + \sum_{s=1}^{5}\sum_{i=1}^{13}\sum_{t=1}^{24} p_s r_i x_{it}^s \\ \text{s.t.} & v_i \geq u_i - u_i' \,,\ v_i \geq 0 \,,\ i=1,\ldots,10 \\ & g_i u_i \leq x_{it}^s \leq G_i u_i \,,\ \forall i,t,s \\ & \sum_{i=1}^{13} x_{it}^s \geq d_t^s \,,\ \forall t,s \\ & x_{it}^s \geq 0 \,,\ \forall i,t,s \\ & u_i \in \{\ell_i,\ldots,n_i\} \,,\ \forall i \end{array}\right\} \qquad (6.8)$$

The model has $10+13+13\times 24\times 5 = 1583$ decision variables (of which 13 are integral and a further 10 automatically integral) and $10+2\times 13\times 24\times 5+24\times 5 = 3250$ constraints.

6.6 Computational Results

The public domain mixed-integer linear programming application LP_SOLVE (see Appendix A.5) was used to find the solutions to the above problems. The expected value problem (6.7) was solved in about 2 seconds while the recourse problem (6.8) was solved in about 40 seconds on a Pentium 166 MHz.

6.6.1 First-Stage Solutions

The solution to the expected value problem (*i.e.* the deterministic solution) is

$EV =\ R\,3,447,117$

i	1	2	3	4	5	6	7	8	9	10	11	12	13
u_i	2	4	8	4	4	0	6	6	6	0	2	0	0

Note that only the optimal values of the u_i are given here. The optimal v_i and x_{it} are listed in Appendix C.1.

The optimal first-stage solutions to the recourse problem (*i.e.* the stochastic solution) is

$RP = \quad R\,3,479,971$

i	1	2	3	4	5	6	7	8	9	10	11	12	13
u_i	2	4	7	4	4	2	6	6	6	0	2	1	0

The optimal second-stage solutions x^s_{it} are listed in Appendix C.2. Under the stochastic solution, there is no unmet demand under any scenario, although gas turbines are used to a *small* extent under scenario 5.

To find the expected result of using the deterministic solution (EEV), we fix the values of $u_1, \ldots, u_{11}$ but not u_{12} and u_{13} since these do not actually form part of the first-stage decision - they are really just indicator variables that indicate how many reserve generating sets may be needed and whether there may be unmet demand. By fixing the above variables and rerunning the recourse program, we obtain $EEV = R\,6,308,938$. The value of the stochastic solution is therefore

$$VSS = EEV - RP = 6,308,938 - 3,479,971 = R\,2,828,967$$

which is a reduction of

$$VSS\,/\,EEV = 2,828,967\,/\,6,308,938 = 44.8\%$$

on the cost of using the deterministic solution. The second-stage variables under the EEV problem are listed in Appendix C.3.

However, this does not give a true indication of the value to Eskom of using the stochastic solution, since Eskom does not actually use the deterministic solution. Eskom's schedules are obtained from a package called COUGER[5], which takes into account all the constraints mentioned above, but assumes that the hourly demands are known with certainty. COUGER uses dynamic programming to obtain a minimum cost solution for a utility over a time horizon of up to one week [1]. It does not, however, produce the deterministic solution as one might expect, since

[5]©1996 ABB Systems Control, a division of ABB Power T&D Co. Inc. http://www.abb.com/americas/usa/tnd_sc

implicit allowance is made for demand uncertainty by allowing the user to define the minimum spinning reserve. COUGER calculates the first-stage solution as

i	1	2	3	4	5	6	7	8	9	10	11	12	13
u_i	2	4	7	4	4	3	6	6	6	0	2	0	0

and has an expected cost of $R3,506,658$. The value of the stochastic solution over this solution is

$$3,506,658 - 3,479,971 = R\,26,687$$

or a percentage reduction of

$$26,687\,/\,3,506,658 = 0.76\%$$

which is small because the solutions are very similar. The only differences between these two solutions are that $u_6 = 2$ and $u_{12} = 1$ in the stochastic solution and $u_6 = 3$ and $u_{12} = 0$ in the COUGER solution. The COUGER solution chooses to keep an extra set on at Lethabo so that gas turbines (which have a high marginal cost) will not be needed under any scenario, nor will there be unmet demand under any scenario. This works out slightly more expensive on average than the recourse solution, due to the high fixed cost of running an extra thermal set. The optimal second-stage solutions for the COUGER solution are listed in Appendix C.4.

6.6.2 Spinning Reserves

A good way to present the second-stage solution is by illustrating the hourly spinning reserves, since the unit commitment problem can be viewed as finding the optimal spinning reserves. On the one hand, the spinning reserve must not be too high since there are high fixed costs and a wastage of resources when units operate at a low output level - perhaps then they should rather not operate at all; while on the other hand, the spinning reserve must not be too low since if demand turns out to be higher than expected, high emergency costs will arise from running expensive reserve generators and, ultimately, a low frequency situation or unmet demand will incur extremely large penalties.

Let y_t^s = spinning reserve during hour t under scenario s

For the purposes of illustration, we define the spinning reserve as the excess capacity in the *thermal* units after the demand has been met, *i.e.*

$$y_t^s = \sum_{i=1}^{11} G_i u_i - \sum_{i=1}^{13} x_{it}^s \tag{6.9}$$

A negative spinning reserve between -342 (= 6 × 57) and 0 (corresponding to the area between the dashed lines in the graphs) implies that gas turbines are used, while a spinning reserve less than -342 (corresponding to the area beneath the dashed lines in the graphs) implies that there is some unmet demand.

Figure 6-3 shows that when the stochastic solution is used, the spinning reserves are positive (and quite large) for all hours of the day under scenarios 1-4, while under scenario 5 (*i.e.* very high demand), the spinning reserve falls just below zero during the daily peak. The stochastic solution leads to spinning reserves that are robust against all possible scenarios.

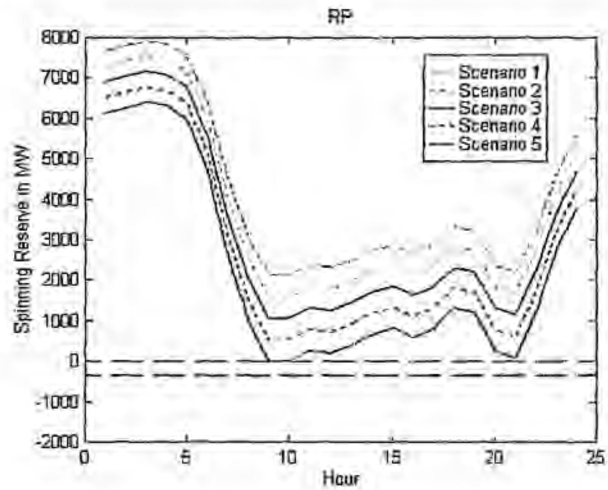

Figure 6-3: Spinning Reserves under the Stochastic Solution

The deterministic solution is calculated under the assumption that demand is equal to the forecast, *i.e.* that scenario 3 occurs. Figure 6-4 shows that this solution leads to a spinning reserve under scenario 3 that is as small as possible yet still positive at the daily peak. (In fact, the spinning reserves are optimal under this scenario). No problems are encountered if scenarios 1 or 2 occur (*i.e.* low or very low demand), since the spinning reserve is always positive. However, if scenarios 4 or 5 occur (*i.e.* high or very high demand), then the spinning reserve becomes large and negative during certain parts of the day, which is very expensive. This is wl makes the deterministic solution so expensive: it is ideal under scenario 3 and good under scenario 1 and 2, but very poor under scenarios 4 and 5. In fact, EEV is so high because it depends on the *arbitrary* and *high* marginal cost of unmet demand of R10,000/MWh. The sensitivities of EEV and VSS with respect to this arbitrarily high marginal cost have not been investigated, since Eskom is interested in the improvement over the COUGER solution rather than the improvement over the deterministic solution which it doesn't use anyway.

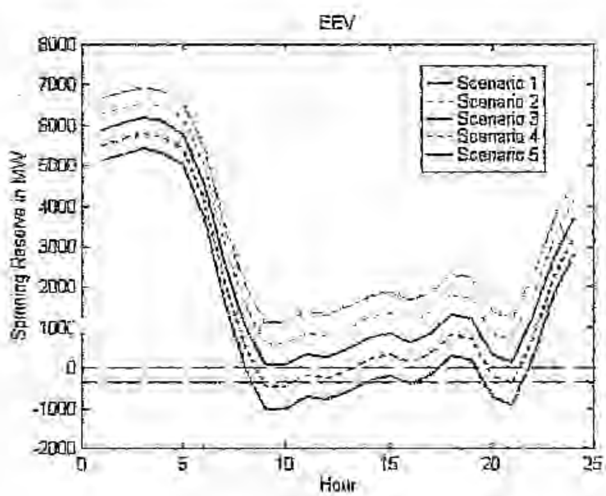

Figure 6-4: Spinning Reserves under the Deterministic Solution

Figure 6-5 shows the spinning reserves under the COUGER solution. When this solution is used, the spinning reserves are large and positive for all hours of the day under all scenarios. This solution is slightly more conservative than the stochastic solution as it avoids all risk of a negative spinning reserve, which implies that reserve generators will never be used. However, as mentioned above, this solution is slightly more expensive to implement on average than the stochastic solution, since an additional thermal set is operated. COUGER does not use the arbitrarily high marginal cost of R10,000/MWh for unmet demand. This is because it does not allow for unmet demand - instead it uses the conservative approach of allowing the user to define a minimum spinning reserve under the expected demand scenario. It is then hoped that the user will choose a sufficiently high minimum value so that under other scenarios, the spinning reserve will not be negative - in fact, this is exactly what happens with the minimum spinning reserve value of 716 MW that Eskom currently uses.

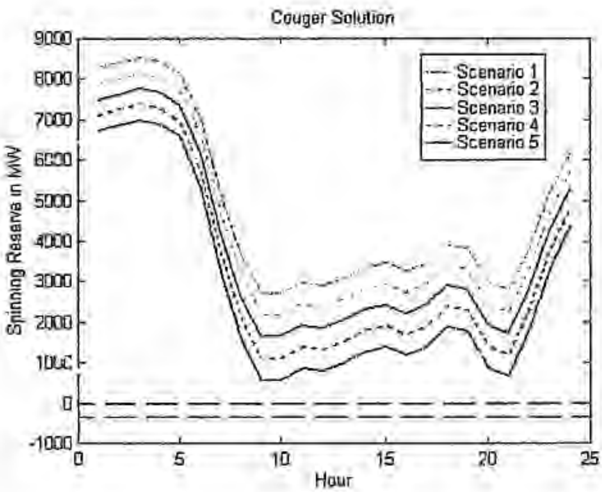

Figure 6-5: Spinning Reserves under the COUGER Solution

6.6.3 Reliability

We are interested in the probability P_1 that the on-line thermal units alone will be unable to meet the demand, in which case the spinning reserve is negative and reserve gas turbines will have to be used. We are also interested in the probability P_2 that there will be some unmet demand. In a sense, P_1 and P_2 could also be called "unreliability" levels. Associated with each

Table 6.5: Unreliability Levels

	C	P_1	P_2
Deterministic Solution	21430	45.157%	22.303%
Stochastic Solution	22426	2.349%	0.431%
COUGER Solution	23019	0.098%	0.009%

solution is the total *capacity* (in MW)

$$C := \sum_{i=1}^{11} G_i u_i \tag{6.10}$$

of the on-line thermal units. It can be seen from Table 6.3 that the demand comes to a *peak* during hours 9 and 10. There will be a negative spinning reserve if and only if the peak demand exceeds the capacity. If we define $\widetilde{\xi}$ as the peak demand, it follows that

$$\widetilde{\xi} \sim N\left(\mu = 21365\,,\, \sigma^2 = (0.025 \times 21365)^2 = 534.125^2\right)$$

since we assume that the standard deviation is 2.5% of the mean, as in Section 6.5 above. Thus for any solution with capacity C, the probability that gas turbines are used (*i.e.* that there is negative spinning reserve) is

$$P_1 = \Pr\left[\widetilde{\xi} \geq C\right] = 1 - \Phi\left(\frac{C - 21365}{534.125}\right) \tag{6.11}$$

and the probability that there is some unmet demand is

$$P_2 = \Pr\left[\widetilde{\xi} \geq C + 342\right] = 1 - \Phi\left(\frac{C - 21023}{534.125}\right) \tag{6.12}$$

Table 6.5 shows these probabilities for the deterministic solution, the stochastic solution and the COUGER solution. Note that as C increases, P_1 and P_2 decrease, so that to minimize the probability of a negative spinning reserve or the probability of unmet demand, we simply maximize the total capacity of the on-line thermal units. However, this approach becomes very expensive in terms of operating cost, so simply minimizing these probabilities is not a sensible criterion in practice.

Note also that when the stochastic solution is used rather than the deterministic solution, there is an increase in total capacity of $\frac{2126-21430}{21430} = 4.65\%$, while the reliability improves by $45.157\% - 2.349\% = 42.808\%$ for P_1 and by 21.872% for P_2. However, when the COUGER solution is used rather than the stochastic solution, the total capacity increases by 2.64%, while the reliabilities P_1 and P_2 only improve by 2.251% and 0.422% respectively. Thus for comparable increases in capacity, the stochastic solution offers a significant improvement in reliability (relative to the deterministic solution), whereas the COUGER solution offers only a marginal improvement in reliability (over the stochastic solution).

6.7 Limitations and Extensions

6.7.1 Decision Frequency

The minimum on-time and minimum off-time of most of Eskom's units is 24 hours, so that the decision-making period of 24 hours is ideal. A couple of the units have a minimum on-time of 18 hours and a minimum off-time of 6 hours. These constraints [illegible]ot be violated if we take decisions every 24 hours. It is difficult to include interruptible lo[illegible] in the model as this is available for at most 2 hours per week.

6.7.2 Generator Failure

This model ignores the possibility that a generating set may become unavailable for use at any time due to mechanical or electrical failure or unscheduled maintenance. Takriti, Birge & Long [45] suggest that it is possible to extend the model to include the possibility of such failure by creating a set of scenarios where the demand is *increased* by the capacity of the failed generating set. The probability of this set of scenarios is equal to the probability that the generating set will fail. Different sets of scenarios can be created for different combinations of generator failures. However, this approach is not necessarily valid, because if the failed generating set were not being used anyway, the cost of its failure (excluding the cost of repair) would be zero, in which case the scenarios should not reflect an increase in demand.

A more valid method suggested in [45] is to approximate the *generating capacity loss* over a period of time. Once again, scenarios are created with increases in demand to reflect the generating capacity loss.

An alternative way to model the problem of generator failure would be to model the n_i as random variables using, say, a *binomial* distribution for each i. Scenarios could be created that included the different possible values of the n_i. However, this would blow up the size of the problem enormously. A *conservative* approach would be to solve the problem with a given number of sets from specific units excluded. In other words, we assume that certain sets will fail.

6.7.3 Nonlinear Formulation

The second-stage problem for a given scenario s and first-stage decision vector $\mathbf{u}$ can be written as

$$Q(\mathbf{u}, s) = \min \quad \sum_{i=1}^{13} \sum_{t=1}^{24} r_i x_{it}^s$$
$$\text{s.t.} \quad \begin{cases} g_i u_i \le x_{it}^s \le G_i u_i \,, \; \forall i, t \\ \sum_{i=1}^{13} x_{it}^s \ge d_t^s \,, \; \forall t \\ x_{it}^s \ge 0 \,, \; \forall i, t \end{cases} \tag{6.13}$$

This is a linear program in $24 \times 13 = 312$ variables, and can be solved by the following heuristic method known as the *economic dispatch routine*:

- Rank all *units* that are on (obtained from the first-stage solution) in increasing order of marginal generating costs r_i.
- Load all units at minimum load.
- Starting with the cheapest unit, increase the loading until either the unit is at maximum loading or the demand is met.
- Continue loading each unit to maximum capacity until total generation equals demand.

The recourse problem can then be rewritten as

$$
\begin{aligned}
\min_{\mathbf{u}} \quad & \sum_{i=1}^{10} h_i \max(u_i - u_i', 0) + 24 \sum_{i=1}^{10} p_i u_i + \sum_{s=1}^{S} p_s Q(\mathbf{u}, s) \\
\text{s.t.} \quad & \ell_i \leq u_i \leq n_i \,, \; u_i \in \mathbb{Z} \,, \text{ for } i = 1, \ldots, 13
\end{aligned}
\tag{6.14}
$$

where S is the total number of scenarios, $\mathbf{u} = (u_1, \ldots, u_{13})^T$ and $Q(\mathbf{u}, s)$ is given by (6.13). Problem (6.14) is a nonlinear integer program in the 13 variables u_i and can be solved by nonlinear programming methods and branching and bounding. It has no constraints on the variables other than simple lower and upper bounds and therefore should be a lot quicker to solve than the mixed-integer LP formulation of the recourse problem which has thousands of variables and constraints. Note that u_{11}, u_{12} and u_{13} do not feature in the first two summations of the objective in (6.14), and feature only implicitly in the third summation. These variables are required to ensure the feasibility of problem (6.13) for all scenarios, and thus affect the objective by ensuring that $Q(\mathbf{u}, s)$ is finite.

6.7.4 Multistage Dynamic Formulation

Eskom is a *going concern* and hence its time horizon is not just one day. It wants to find a way of optimally scheduling its generating units so that costs will be minimized over a much longer period of time, such as a year. The problem with the one-day model is that the cost is minimized over the day with no regard to the final state that the system will end up in. In this model the state is the vector $\mathbf{u}$ (the number of sets on at each unit) and can change only once from day to day. The short-term time horizon of one day implies that merely applying the same one-day model over a number of days does not necessarily lead to the cheapest schedule over the entire period, since each day is considered independently. As the final state of one day will be the initial state of the next day, a multistage dynamic program can be formulated with each stage being a period of one day, and the state being the vector $\mathbf{u}$. The dynamic program would involve solving a recourse problem for each possible state at every stage. Such a dynamic program may well be tractable, particularly if the nonlinear formulation is used to solve each recourse problem.

A massive simplification would occur if we assumed that demand had the same distribution on *all days*, independently. We could then obtain a ***steady-state solution*** by solving the one-day problem defined by the first-stage variables $u_i = u_i' \ \forall i$ where the u_i' would no longer be constants, but decision variables. However, this approach fails in practice since the demand on weekdays differs significantly from the demand on weekends. This difficulty can be overcome by creating a steady-state multistage problem over a period of one week, assuming that weekly demand patterns do not change from week to week. In this formulation, the u_i at the start of the week would be the same as the u_i at the start of the next week. Decisions on unit commitment would be made at the start of each day, as in the one-day model. Such a model would involve eight stages (one for each day of the week from Monday through to Monday) and may well be tractable since once the u_i decisions have been made, the x_{it}^s decisions follow automatically according to the economic dispatch routine of the previous section.

Appendix A

Deterministic Mathematical Programming

This appendix provides a background in the theory and methodology of deterministic mathematical programming, including linear programming, nonlinear programming and integer programming. The appendix starts with mathematical background (chiefly from convex analysis) and ends with a section on computer software for mathematical programming.

A.1 Convex Analysis

This section provides definitions from convex analysis and set topology that are frequently used in mathematical programming. Further mathematical background and proofs of unproved claims can be found in Marlow [35] or Lay [31].

A.1.1 Topological Concepts

A *neighbourhood* of $\mathbf{x}$ of radius $\delta > 0$ is a set of the form $\{\mathbf{y} \mid \|\mathbf{y} - \mathbf{x}\| < \delta\}$. Such a set is also known as an *open ball* of radius δ centred at $\mathbf{x}$. A set $S \subset \mathbb{R}^n$ is an *open* set if each of its elements has a neighbourhood that lies entirely within the set. The *interior* of a set $S \subset \mathbb{R}^n$ is the union of all the open sets contained in S. A set $S \subset \mathbb{R}^n$ is a *closed* set if its complement $S^c = \mathbb{R}^n - S$ is an open set. The *closure* of any set $S \subset \mathbb{R}^n$ is the smallest closed set containing S. The *boundary* of a set is that part of the closure that is not in the interior. A set $S \subset \mathbb{R}^n$ is *bounded* if it is contained within a sphere of finite radius. A set is *compact* if it is both closed and bounded.

A.1.2 Concepts of Convexity

A set is convex if the line segment joining any two of its elements lies entirely within the set.

Definition 20 (Convex Set) *A set $C \subset \mathbb{R}^n$ is said to be a convex set if for every $\mathbf{x}^1, \mathbf{x}^2 \in C$ and every $\lambda \in (0,1)$, the point $\lambda\mathbf{x}^1 + (1-\lambda)\mathbf{x}^2 \in C$.*

The *intersection* of convex sets is a convex set, while the *union* of convex sets is not, in general, a convex set.

Definition 21 (Convex Function) *A function $f : D \subset \mathbb{R}^n \to \mathbb{R}$, where D is a convex set, is a convex function if*

$$f\left(\lambda\mathbf{x}^1 + (1-\lambda)\mathbf{x}^2\right) \leq \lambda f\left(\mathbf{x}^1\right) + (1-\lambda) f\left(\mathbf{x}^2\right)$$

for every $\mathbf{x}^1, \mathbf{x}^2 \in D$ and every $\lambda \in (0,1)$.

The *sum* of two convex functions is a convex function, and a convex function multiplied by a positive constant is a convex function.

A function f defined on a convex set D is said to be a *concave function* if $-f$ is a convex function. A *linear function* is a function of the form $f(\mathbf{x}) = \mathbf{b}^T\mathbf{x}$, while an *affine function* is a function of the form $f(\mathbf{x}) = a + \mathbf{b}^T\mathbf{x}$, for some $a \in \mathbb{R}$ and $\mathbf{b} \in \mathbb{R}^n$. Linear functions and affine functions are convex as well as concave.

A *convex combination* of the set $X = \{\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^r\} \subset \mathbb{R}^n$ is a linear combination $\mathbf{x} = \sum_{i=1}^{r} \lambda_i \mathbf{x}^i$, where $\sum_{i=1}^{r} \lambda_i = 1$ and $\lambda_i \geq 0 \; \forall i$.

Definition 22 (Convex Hull) *The convex hull of the set of points $X = \{\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^r\} \subset \mathbb{R}^n$, denoted* $\operatorname{conv}\{\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^r\}$, *is the union of X and the set of all convex combinations of* *points in X, i.e.*

$$\operatorname{conv}\{\mathbf{x}^1,\mathbf{x}^2,\ldots,\mathbf{x}^r\} = \left\{\mathbf{x} \mid \mathbf{x} = \sum_{i=1}^{r} \lambda_i \mathbf{x}^i,\ \sum_{i=1}^{r} \lambda_i = 1,\ \lambda_i \geq 0\ \forall i\right\} \tag{A.1}$$

The convex hull of a finite number of points is called a *convex polyhedron* or a *bounded convex polyhedral set.* A *vertex* of a convex polyhedron $\mathcal{P}$ is a point $\hat{\mathbf{x}} \in \mathcal{P}$ such that the line segment connecting any two points in $\mathcal{P}$, both different from $\hat{\mathbf{x}}$, does not contain $\hat{\mathbf{x}}$. Formally, $\hat{\mathbf{x}}$ is a vertex of $\mathcal{P}$ if

$$\nexists\, \mathbf{y}, \mathbf{z} \in \mathcal{P},\ \mathbf{y} \neq \hat{\mathbf{x}} \neq \mathbf{z},\ \lambda \in (0,1),\ \text{such that } \hat{\mathbf{x}} = \lambda \mathbf{y} + (1-\lambda)\mathbf{z}$$

A set C is a *cone* if $\mathbf{x} \in C$ implies $\lambda \mathbf{x} \in C$ for all $\lambda \geq 0$. A set C is a *convex cone* if for any two elements $\mathbf{y}, \mathbf{z} \in C$, it follows that $\lambda \mathbf{y} + \mu \mathbf{z} \in C\ \forall \lambda, \mu \geq 0$. A *positive combination* of the set $Y = \{\mathbf{y}^1, \mathbf{y}^2, \ldots, \mathbf{y}^r\} \subset \mathbb{R}^n$ is a linear combination $\mathbf{y} = \sum_{i=1}^{r} \lambda_i \mathbf{y}^i$, where $\lambda_i \geq 0\ \forall i$.

Definition 23 (Positive Hull) *The positive hull of the set of points $Y = \{\mathbf{y}^1, \mathbf{y}^2, \ldots, \mathbf{y}^r\} \subset \mathbb{R}^n$, denoted $\operatorname{pos}\{\mathbf{y}^1, \mathbf{y}^2, \ldots, \mathbf{y}^r\}$, is the union of Y and the set of all positive combinations of points in Y, i.e.*

$$\operatorname{pos}\{\mathbf{y}^1,\mathbf{y}^2,\ldots,\mathbf{y}^r\} = \left\{\mathbf{y} \mid \mathbf{y} = \sum_{i=1}^{r} \lambda_i \mathbf{y}^i,\ \lambda_i \geq 0\ \forall i\right\} \tag{A.2}$$

The positive hull of a finite number of points is called a *convex polyhedral cone.* Any element of a convex polyhedral cone can be represented as a positive combination of its *generating elements* $\{\mathbf{y}^1, \mathbf{y}^2, \ldots, \mathbf{y}^r\}$. The *algebraic sum* of the sets P and C is the set

$$\{\mathbf{z} \mid \mathbf{z} = \mathbf{x} + \mathbf{y},\ \mathbf{x} \in \mathcal{P},\ \mathbf{y} \subset \mathcal{C}\}$$

A *convex polyhedral set* is the algebraic sum of a convex polyhedron and a convex polyhedral cone. The sides of a convex polyhedral set are called *facets.*

A.2 Linear Programming

The *standard form* of a linear program (LP) was given in (1.2) as

$$\min \left\{ \mathbf{c}^T \mathbf{x} \mid A\mathbf{x} = \mathbf{b}, \mathbf{x} \geq 0 \right\} \tag{A.3}$$

where the vectors $\mathbf{c} \in \mathbb{R}^n$, $\mathbf{b} \in \mathbb{R}^m$ and the $m \times n$ matrix A are given and the decision vector $\mathbf{x} \in \mathbb{R}^n$ is to be determined. It was explained in Section 1.1.1 that any linear program can be converted into the standard form (A.3). Linear programming is a well-developed field and there are many good references on the subject. For a detailed presentation of linear programming, see Dantzig [14] or Luenberger [34]. This section gives a review of linear programming based on Section 2.9 of Birge & Louveaux [10] and Section 1.6 of Kall & Wallace [28].

A.2.1 Feasible Sets and Solutions

We assume that $m \leq n$ and $\text{rank}(A) = m$. A *solution* to (A.3) is a vector $\mathbf{x}$ that satisfies $A\mathbf{x} = \mathbf{b}$. A *feasible solution* is a solution $\mathbf{x}$ with $\mathbf{x} \geq 0$. We define the *feasible set* as

$$\mathcal{B} := \{\mathbf{x} \mid A\mathbf{x} = \mathbf{b}, \mathbf{x} \geq 0\} \tag{A.4}$$

An *optimal solution* $\mathbf{x}^*$ is a feasible solution such that $\mathbf{c}^T\mathbf{x}^* \leq \mathbf{c}^T\mathbf{x}$ for all feasible solutions $\mathbf{x} \in \mathcal{B}$. A *basis* is a choice of m linearly independent columns of A. Associated with a basis is a nonsingular $m \times m$ submatrix B of the corresponding columns, so that after a suitable rearrangement, A can be partitioned into $A = [B, N]$. We also partition $\mathbf{x}^T = [\mathbf{x}_B^T, \mathbf{x}_N^T]$ and $\mathbf{c}^T = [\mathbf{c}_B^T, \mathbf{c}_N^T]$ to correspond to the basic columns B and nonbasic columns N of A. The system of equations $A\mathbf{x} = \mathbf{b}$ can then be rewritten as

$$B\mathbf{x}_B + N\mathbf{x}_N = \mathbf{b} \tag{A.5}$$

or equivalently as

$$\mathbf{x}_B = B^{-1}\mathbf{b} - B^{-1}N\mathbf{x}_N \tag{A.6}$$

If all $n-m$ components of $\mathbf{x}$ not associated with the columns of B are set equal to zero, *i.e.* $\mathbf{x}_N = \mathbf{0}$, then the solution to the resulting set of equations is said to be a *basic solution* with respect to the basis B. The basic solution is

$$\left.\begin{aligned} \mathbf{x}_B &= B^{-1}\mathbf{b} \\ \mathbf{x}_N &= \mathbf{0} \\ z &= \mathbf{c}_B^T B^{-1}\mathbf{b} \end{aligned}\right\} \tag{A.7}$$

and if the condition $B^{-1}\mathbf{b} \geq \mathbf{0}$ is satisfied, then the solution is said to be a *basic feasible solution*. A basis is said to be feasible (or optimal) if its associated basic solution is feasible (or optimal). A basic feasible solution is called *nondegenerate* if $B^{-1}\mathbf{b} > \mathbf{0}$ and *degenerate* if $B^{-1}\mathbf{b} \not> \mathbf{0}$.

Basic feasible solutions play a dominant role in describing feasible sets of linear programs. We denote the basic feasible solutions as $\mathbf{x}^{(i)}$, $i = 1, \ldots, r$. It is shown in Section 1.6.1 of Kall & Wallace [28] that

- If $\mathcal{B} \neq \emptyset$ then there exists at least one basic feasible solution.
- If $\mathcal{B} \neq \emptyset$ and $\mathcal{B}$ is a bounded set, then $\mathcal{B}$ is the convex hull of the set of its basic feasible solutions, *i.e.* $\mathcal{B} = \operatorname{conv}\{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(r)}\}$, and the basic feasible solutions coincide with the vertices of $\mathcal{B}$.
- The set $\mathcal{C} := \{\mathbf{y} \mid A\mathbf{y} = \mathbf{0}, \mathbf{y} \geq \mathbf{0}\}$ is a convex polyhedral cone.
- The feasible set $\mathcal{B} \neq \emptyset$ is bounded iff $\mathcal{C} = \{\mathbf{0}\}$.
- The feasible set $\mathcal{B}$ is the algebraic sum of the convex polyhedron $\mathcal{P} := \operatorname{conv}\{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(r)}\}$ and the convex polyhedral cone $\mathcal{C}$, and hence $\mathcal{B}$ is a convex polyhedral set. This means that every $\hat{\mathbf{x}} \in \mathcal{B}$ can be represented as $\hat{\mathbf{x}} = \hat{\mathbf{z}} + \hat{\mathbf{y}}$, where $\hat{\mathbf{z}} \in \mathcal{P}$ and $\hat{\mathbf{y}} \in \mathcal{C}$. Formally, we write $\mathcal{B} = \mathcal{P} + \mathcal{C}$.
- The linear program (A.3) is solvable iff $\mathcal{B} \neq \emptyset$ and $\mathbf{c}^T\mathbf{y} \geq 0 \; \forall \mathbf{y} \in \mathcal{C}$. Given that these two conditions are satisfied, there is at least one basic feasible solution that is an optimal solution.

The optimal solution to a linear program is not necessarily unique. There may be more than one basic feasible solution that is optimal. In this case, any convex combination of the optimal basic feasible solutions will also be an optimal solution. The case of degeneracy is an example of a situation where the optimal solution is not unique.

A.2.2 The Simplex Method

The above results imply that to solve a linear program, it is sufficient to consider only basic solutions as candidates for the optimal solution. There are a finite number of basic solutions and therefore an optimal solution can be obtained in a finite number of steps. The idea of the *(primal) simplex method* is to proceed from one basic feasible solution to another, in such a way as to decrease the value of the objective function each time, until a minimum is reached.

Starting with a basic feasible solution, the next basic feasible solution is obtained by selecting an *entering* variable (any nonbasic variable whose increase leads to a decrease in the objective value) and a *leaving* variable (the first to become negative as the entering variable increases). The entering variable is substituted for the leaving variable by *pivoting*. An optimal solution is reached when no entering variable can be found.

A linear program is *unbounded* if an entering variable exists for which no leaving variable can be found. In some cases, a basic feasible solution is not immediately available. The *two-phase* simplex method is then implemented. *Phase one* uses *artificial* variables to find a basic feasible solution (if one exists) by minimizing the sum of the artificial variables. If a solution exists with the sum of the artificial variables equal to zero, then the original problem is feasible and *phase two* continues with the original objective function, starting with the basic feasible solution given by phase one. If the optimal objective of the phase one problem is positive, then the original problem is *infeasible*. It is shown in Section 1.6.2 of Kall & Wallace [28] that

- A basic feasible solution is optimal iff $\mathbf{c}_N^T - \mathbf{c}_B^T B^{-1} N \geq \mathbf{0}^T$.
- Under the assumption of nondegeneracy of the basic feasible solutions, the simplex method yields after finitely many steps either an optimal solution or the information that the program is infeasible or unbounded.

A.2.3 Duality

The concept of duality was used in developing the L-shaped algorithm in Section 3.5.3. Given the *primal* program (A.3), the *dual program* is formulated as

$$\max \{\mathbf{b}^T \mathbf{u} \mid A^T \mathbf{u} \leq \mathbf{c}\} \tag{A.8}$$

The dual of any linear program can be found by writing the program in the standard form (A.3) and then converting to the dual program (A.8). The dual of the dual program is the primal program. The variables $\mathbf{u}$ are called *dual variables* and are also known as *dual prices, shadow prices* or *multipliers*. One dual variable is associated with each constraint of the primal program. The dual variable u_i can be interpreted as the marginal value of an increase in resource b_i. When the primal constraint is an equality, the dual variable is free.

We define the feasible set of the dual program as

$$\mathcal{D} := \{\mathbf{u} \mid A^T \mathbf{u} \leq \mathbf{c}\} \tag{A.9}$$

and adopt the convention that

$$\left.\begin{array}{ll} \min\limits_{\mathbf{x} \in \mathcal{B}} \mathbf{c}^T \mathbf{x} = \infty & \text{if } \mathcal{B} = \emptyset \\ \max\limits_{\mathbf{u} \in \mathcal{D}} \mathbf{b}^T \mathbf{u} = -\infty & \text{if } \mathcal{D} = \emptyset \end{array}\right\} \tag{A.10}$$

The *weak duality theorem* of linear programming states that

$$\mathbf{c}^T \mathbf{x} \geq \mathbf{b}^T \mathbf{u}, \ \forall \mathbf{x} \in \mathcal{B}, \mathbf{u} \in \mathcal{D} \tag{A.11}$$

and hence $\min\limits_{\mathbf{x} \in \mathcal{B}} \mathbf{c}^T \mathbf{x} \geq \max\limits_{\mathbf{u} \in \mathcal{D}} \mathbf{b}^T \mathbf{u}$. The *strong duality theorem* of linear programming states that if either $\mathcal{B} \neq \emptyset$ or $\mathcal{D} \neq \emptyset$ (*i.e.* either the primal program or the dual program is feasible) then it follows that

$$\min_{\mathbf{x} \in \mathcal{B}} \mathbf{c}^T \mathbf{x} = \max_{\mathbf{u} \in \mathcal{D}} \mathbf{b}^T \mathbf{u} \tag{A.12}$$

If the dual problem is unbounded, then the primal problem is infeasible. Similarly, if the primal problem is unbounded, the dual problem is infeasible. It is also possible for both problems to be infeasible simultaneously. The primal program has an optimal solution $\mathbf{x}^*$ if and only if the dual program has an optimal solution $\mathbf{u}^*$, and when this happens, $\mathbf{c}^T\mathbf{x}^* = \mathbf{b}^T\mathbf{u}^*$ and the primal and dual solutions satisfy the *complementary slackness* conditions:

$$\left.\begin{aligned} (A_{i\cdot}\mathbf{x}^* - b_i)\, u_i^* = 0 \quad & \text{for } i = 1,\ldots,m \\ (A_{\cdot j}^T\mathbf{u}^* - c_j)x_j^* = 0 \quad & \text{for } j = 1,\ldots,n \end{aligned}\right\} \tag{A.13}$$

where $A_{i\cdot}$ is the ith row of A and $A_{\cdot j}$ is the jth column of A. The conditions (A.13) imply that $A_{i\cdot}\mathbf{x}^* = b_i$ or $u_i^* = 0$ or both, for $i = 1,\ldots,m$, and that $A_{\cdot j}^T\mathbf{u}^* = c_j$ or $x_j^* = 0$ or both, for $j = 1,\ldots,n$. Thus the optimal solution of the dual problem can be recovered from the optimal solution of the primal [illegible] vice versa.

The *dual simplex m*[illegible] primal problem what the iterations of the simplex method would be o[illegible] finds the leaving variable (one that is strictly negative), then the entering va[illegible] that would become negative in the objective line). The dual simplex m[illegible] ion to the primal problem that is optimal but infeasible, and works towards feasibility. [illegible]is contrasts with the primal simplex method, which starts with a feasible solution and works towards optimality.

A.3 Nonlinear Programming

Nonlinear programming, like linear programming, is a well-developed field and there are many good references on the subject, such as Luenberger [34]. For an introductory text, see Taha [44]. This section briefly reviews aspects of nonlinear programming that are relevant to stochastic programming.

A.3.1 Standard Form

The general form of a mathematical program was stated in (1.3) as

$$\min\{g_0(\mathbf{x}) \mid g_i(\mathbf{x}) \le 0\,,\ i = 1,\ldots,m\,,\ \mathbf{x} \in X \subseteq \mathbb{R}^n\}$$

where the functions g_i, $i = 0, \ldots, m$ are given. In this section, we use a slightly different *standard form* for the *nonlinear program* (NLP), *viz.*

$$\left.\begin{array}{ll} \min\limits_{\mathbf{x}\in X} & f(\mathbf{x}) \\ \text{s.t.} & g_i(\mathbf{x}) \leq 0\,,\ i = 1, \ldots, m \\ & h_j(\mathbf{x}) = 0\,,\ j = 1, \ldots, p \end{array}\right\} \tag{A.14}$$

or equivalently, with $\mathbf{g}(\mathbf{x}) := (g_1(\mathbf{x}), \ldots, g_m(\mathbf{x}))^T$ and $\mathbf{h}(\mathbf{x}) := (h_1(\mathbf{x}), \ldots, h_p(\mathbf{x}))^T$,

$$\min\{f(\mathbf{x}) \mid \mathbf{g}(\mathbf{x}) \leq \mathbf{0}\,,\ \mathbf{h}(\mathbf{x}) = \mathbf{0}\,,\ \mathbf{x} \in X\} \tag{A.15}$$

Once again, we assume that the functions $f : \mathbb{R}^n \to \mathbb{R}$, $\mathbf{g} : \mathbb{R}^n \to \mathbb{R}^m$, $\mathbf{h} : \mathbb{R}^n \to \mathbb{R}^p$ are given, and that we must determine the decision vector $\mathbf{x} \in \mathbb{R}^n$. In this overview, we assume (unless stated otherwise) that X is a convex set, the functions f and g_i, $i = 1, \ldots, m$ are convex in $\mathbf{x}$ and the functions h_j, $j = 1, \ldots, p$ are affine in $\mathbf{x}$, so that (A.14) is a convex program according to Proposition 1 and Corollary 1 of Section 1.1.2. Furthermore, we assume that the functions f, $\mathbf{g}$ and $\mathbf{h}$ have continuous partial derivatives, so that (A.15) is a *smooth* convex program.

An inequality constraint $g_i(\mathbf{x}) \leq 0$ is said to be *active* at $\mathbf{x}$ if $g_i(\mathbf{x}) = 0$ and *inactive* at $\mathbf{x}$ if $g_i(\mathbf{x}) < 0$.

A.3.2 Kuhn-Tucker Conditions

We convert each inequality constraint $g_i(\mathbf{x}) \leq 0$ into the equality constraint $g_i(\mathbf{x}) + s_i^2 = 0$ by adding the non-negative slack variable $s_i^2 \geq 0$ for $i = 1, \ldots, m$, and define $\mathbf{s} = (s_1, \ldots, s_m)^T$ and $\mathbf{s}^2 = (s_1^2, \ldots, s_m^2)^T$. We then create the *Lagrangian function*

$$\mathcal{L}(\mathbf{x}, \mathbf{s}, \boldsymbol{\lambda}, \boldsymbol{\pi}) = f(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i \left(g_i(\mathbf{x}) + s_i^2\right) + \sum_{j=1}^{p} \pi_j h_j(\mathbf{x}) \tag{A.16}$$

where $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_m)^T$ is the vector of *Lagrange multipliers* associated with the inequality constraints and $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_p)^T$ is the vector of Lagrange multipliers associated with the equality constraints. At all points $(\mathbf{x}, \mathbf{s})$ where $\mathcal{L}$ is stationary with respect to $\boldsymbol{\lambda}$ and $\boldsymbol{\pi}$,

$$\frac{\partial \mathcal{L}}{\partial \lambda_i} = g_i(\mathbf{x}) + s_i^2 = 0 \Rightarrow g_i(\mathbf{x}) \leq 0\,, \text{ for } i = 1, \ldots, m \tag{A.17}$$

and

$$\frac{\partial \mathcal{L}}{\partial \pi_j} = h_j(\mathbf{x}) = 0\,, \text{ for } j = 1, \ldots, p \tag{A.18}$$

and therefore any point $\mathbf{x}$ that is a stationary point of $\mathcal{L}$ with respect to $\boldsymbol{\lambda}$ and $\boldsymbol{\pi}$ is a feasible point. Note that:

- The Lagrangian function is convex in $\mathbf{x}$ if $\boldsymbol{\lambda} \geq \mathbf{0}$ and $\boldsymbol{\pi}$ is unrestricted in sign, since we assume that the functions f and $\mathbf{g}$ are convex and $\mathbf{h}$ is affine.
- The Lagrangian function is convex in $\mathbf{s}$ iff $\boldsymbol{\lambda} \geq \mathbf{0}$.
- For any feasible point $\mathbf{x} \in \mathcal{B} := \{\mathbf{x} \mid \mathbf{g}(\mathbf{x}) \leq \mathbf{0}\,, \mathbf{h}(\mathbf{x}) = \mathbf{0}\}$, the Lagrangian function reduces to $\mathcal{L}(\mathbf{x}, \mathbf{s}, \boldsymbol{\lambda}, \boldsymbol{\pi}) = f(\mathbf{x})\ \forall \boldsymbol{\lambda}, \boldsymbol{\pi}$.

Solving the program (A.15) with respect to $\mathbf{x}$ is then equivalent to minimizing the Lagrangian function (A.16) with respect to $\mathbf{x}$ and $\mathbf{s}$, subject to the Lagrangian function being stationary with respect to $\boldsymbol{\lambda}$ and $\boldsymbol{\pi}$. At a minimum, it is necessary that

$$\frac{\partial \mathcal{L}}{\partial \mathbf{x}} = \nabla f(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i \nabla g_i(\mathbf{x}) + \sum_{j=1}^{p} \pi_j \nabla h_j(\mathbf{x}) = 0 \tag{A.19}$$

and

$$\frac{\partial \mathcal{L}}{\partial s_i} = 2\lambda_i s_i = 0\,, \text{ for } i = 1, \ldots, m \tag{A.20}$$

Provided that $\boldsymbol{\lambda} \geq \mathbf{0}$, the conditions (A.19) and (A.20) are also sufficient for the point $(\mathbf{x}, \mathbf{s})$ to be a minimum of (A.15) because then $\mathcal{L}$ is convex in $\mathbf{x}$ and $\mathbf{s}$, for fixed $\boldsymbol{\lambda}$, $\boldsymbol{\pi}$. The equations (A.20) reveal that:

- If $\lambda_i > 0$ then $s_i = 0$ (and hence $g_i(\mathbf{x}) = 0$). This means that the corresponding resource is scarce, and consequently it is exhausted completely and the constraint is active.
- If $s_i^2 > 0$ (and $g_i(\mathbf{x}) < 0$) then $\lambda_i = 0$. This means that the corresponding resource is not scarce, and consequently the constraint is inactive and does not affect the value of f.

Conditions (A.17) and (A.20) thus imply that $\lambda_i g_i(\mathbf{x}) = 0 \; \forall i$, or equivalently,

$$\sum_{i=1}^{m} \lambda_i g_i(\mathbf{x}) = 0$$

since $\lambda_i \geq 0 \; \forall i$ and $g_i(\mathbf{x}) \leq 0 \; \forall i$. It is shown in Section 1.7.1 of Kall & Wallace [28] that if the *regularity condition*

$$\exists \hat{\mathbf{x}} \text{ s.t. } g_i(\hat{\mathbf{x}}) < 0, \; i = 1, \ldots, m \tag{A.21}$$

(which is called the *Slater condition*) holds, then the feasible point $\mathbf{x}^* \in B$ is a global optimum of (A.15) if and only if there exists $\boldsymbol{\lambda}^* \geq \mathbf{0}$, $\boldsymbol{\pi}^*$ such that the *Kuhn-Tucker conditions*

$$\nabla f(\mathbf{x}^*) + \sum_{i=1}^{m} \lambda_i^* \nabla g_i(\mathbf{x}^*) + \sum_{j=1}^{p} \pi_j^* \nabla h_j(\mathbf{x}^*) = 0 \tag{A.22a}$$

$$\sum_{i=1}^{m} \lambda_i^* g_i(\mathbf{x}^*) = 0 \tag{A.22b}$$

are satisfied at $\mathbf{x}^*$. If the Slater condition is not satisfied (*i.e.* there is no feasible point that lies strictly within the interior of B), then the Kuhn-Tucker conditions do not necessarily hold at an optimum.

It is also shown in Section 1.7.1 of Kall & Wallace [28] that when a more general regularity condition holds, the Kuhn-Tucker conditions are necessary, but not sufficient, for optimality of a nonlinear program that is not necessarily convex. Second-order necessary and sufficient optimality conditions are derived in Section 10.6 of Luenberger [34].

A.3.3 Solution Methods in Nonlinear Programming

Unlike linear programs, nonlinear programs cannot generally be solved in a finite number of steps. This is because an infinite number of potential solutions must be considered. Most nonlinear programming algorithms involve *iterative* procedures that aim to converge to a solution of the nonlinear program.

Primal methods of solution work on the original problem directly by searching through the feasible region for the optimal solution. The solution at each iteration is feasible and the value of the objective decreases at each iteration. If the procedure is terminated early, the solution obtained will always be feasible and may be close to an optimal solution. Most primal methods are not dependent on a special problem structure, such as convexity. However, it can be difficult to find an initial feasible solution. *Descent direction* methods are primal methods that take steps through the feasible region of the form

$$x^{(\nu+1)} = x^{(\nu)} + \alpha^{(\nu)} d^{(\nu)}$$

where $x^{(\nu)}$ is the solution at the νth iteration, $d^{(\nu)}$ is a *direction* vector and $\alpha^{(\nu)}$ is a non-negative scalar that is chosen to minimize $f\left(x^{(\nu+1)}\right)$ subject to $x^{(\nu+1)}$ being feasible. Examples of descent methods include the *feasible direction* method, the *gradient projection* method and the *reduced gradient* method. See Chapter 11 of Luenberger [34] for details on primal methods.

Penalty methods and *barrier* methods approximate constrained problems of the form (A.14) by unconstrained problems (*i.e.* problems with an objective but no constraints). Penalty methods add a term to the objective function that allocates a high cost for violation of the constraints, while barrier methods add a term to the objective that favours interior points of the feasible region as opposed to points near the boundary. See Chapter 12 in Luenberger [34] for details on penalty and barrier methods.

Cutting plane methods can be used to solve convex programs. These algorithms replace the original problem by a series of ever-improving approximating linear programs. The approximating linear programs represent the original problem more and more accurately so that the solutions of these linear programs converge to the solution of the original problem.

Note that if the optimal Lagrange multipliers λ^* and π^* in (A.16) were known, solving the non-linear program would be relatively easy, as we would simply have to solve the equations (A.19) and (A.20). *Dual methods* tackle an alternate problem, the dual problem, whose unknowns are the Lagrange multipliers of the original problem. See Chapter 13 of Luenberger [34] for details on cutting plane methods and dual methods.

The *augmented Lagrangian method* combines a Lagrangian approach with a penalty method to create an unconstrained problem with a modified objective function - see Section 1.7.4 in Kall & Wallace [28].

Suppose that, in solving a nonlinear program, we knew which constraints were active at the optimal solution, and which were inactive. We would then only need to include the active constraints in our formulation and solve the problem as a nonlinear program subject to equality constraints (the active constraints). The inactive constraints can be ignored since they do not affect the solution - they are satisfied without being enforced. The concept of identification of the set of active constraints is used in sequential quadratic programming methods, amongst others.

Sequential quadratic programming (SQP) methods have been shown to be highly efficient and accurate methods of nonlinear programming [40]. These methods approximate the Lagrangian function of the original problem by a quadratic function. At each major iteration, an approximation is made of the Hessian matrix of second derivatives of the Lagrangian function. This is then used to generate a quadratic programming subproblem whose solution is used to form a search direction for a line search procedure. The quadratic subproblem is obtained by linearizing the nonlinear constraints and can be solved by efficient methods of quadratic programming. See the MATLAB Optimization Toolbox User Guide [36] or Gill *et al.* [24] for details on SQP methods.

Sometimes nonlinear programs arise in which the functions are not differentiable or have discontinuous derivatives. Optimality conditions must then be written in terms of *subgradients* and *subdifferentials* and methods of *non-differentiable* or *non-smooth optimization* must be used - see Lemaréchal [32]. A difficulty in stochastic programming is that discrete distributions often lead to a non-differentiable expected recourse function, and thus standard methods of nonlinear programming cannot be used. This is one of the reasons why special algorithms are developed to solve stochastic programs.

A.4 Integer Programming

The phrase *integer programming* normally refers to integer *linear* programming, i.e. linear programming where some of the variables are restricted to be integers. A deterministic integer programming problem can be formulated as

$$\left.\begin{array}{ll}\min & \mathbf{c}^T\mathbf{x} \\ \text{s.t.} & A\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \geq 0 \\ & x_i \in \mathbb{Z},\ \forall i \in I\end{array}\right\} \quad \text{(A.23)}$$

where $\mathbb{Z}$ is the set of integers and I is the set of subscripts of variables with integer restrictions. More general integer nonlinear programs can obviously be formulated. This section gives a brief overview of two fundamental methods in integer linear programming, *viz.* branch-and-bound and cutting plane methods. The development in this section is based on Section 3.7 in Kall & Wallace [28]. For further information on integer programming, see Taha [43].

A.4.1 Branch and Bound

To simplify the presentation, we assume that all the variables have integer restrictions of the form $x_i \in \{a_i, a_i+1, \ldots, b_i\}$ where $\{a_i, \ldots, b_i\}$ is the set of all integers from a_i to b_i. The branch and bound procedure is based on replacing $x_i \in \{a_i, \ldots, b_i\}$ by $a_i \leq x_i \leq b_i\ \forall i$ and solving the corresponding *relaxed linear program* to obtain the optimal solution $\hat{\mathbf{x}}$. If $\hat{\mathbf{x}}$ happens to be integral, we have the optimal integer solution, since integrality occurs without being enforced. If $\hat{\mathbf{x}}$ is not integral, we have obtained a lower bound $\underline{z} = \mathbf{c}^T\hat{\mathbf{x}}$ on the true optimal objective value, since when a minimization problem becomes less constrained, the optimal objective cannot be higher than it originally was.

If $\hat{\mathbf{x}}$ is not integral, we continue by picking the branching variable x_j and an integer d_j. Normally we choose x_j as a variable that was non-integral in the relaxed LP solution and choose $d_j = \lfloor x_j \rfloor$. We then branch by replacing the original problem

$$\left.\begin{array}{ll}\min & \mathbf{c}^T\mathbf{x} \\ \text{s.t.} & A\mathbf{x} = \mathbf{b} \\ & x_i \in \{a_i, \ldots, b_i\}\ \forall i\end{array}\right\} \quad \text{(A.24)}$$

by the two problems

$$\left.\begin{array}{ll} \min & \mathbf{c}^T\mathbf{x} \\ \text{s.t.} & A\mathbf{x} = \mathbf{b} \\ & x_i \in \{a_i, \ldots, b_i\} \ \forall i \neq j \\ & x_j \in \{a_j, \ldots, d_j\} \end{array}\right\} \tag{A.25}$$

and

$$\left.\begin{array}{ll} \min & \mathbf{c}^T\mathbf{x} \\ \text{s.t.} & A\mathbf{x} = \mathbf{b} \\ & x_i \in \{a_i, \ldots, b_i\} \ \forall i \neq j \\ & x_j \in \{d_{j+1}, \ldots, b_i\} \end{array}\right\} \tag{A.26}$$

Each time we branch, we replace the original problem by two similar problems that each investigate their own part of the solution space. The idea behind *branching* is to continue to branch until it is no longer necessary to branch, and the branch is *fathomed.* When we solve problems (A.25) and (A.26), we relax the integrality constraints to obtain the relaxed linear programs

$$\min\left\{\mathbf{c}^T\mathbf{x} \mid A\mathbf{x} = \mathbf{b}\,,\, a_i \leq x_i \leq b_i \ \forall i \neq j\,,\, a_j \leq x_j \leq d_j\right\} \tag{A.27}$$

and

$$\min\left\{\mathbf{c}^T\mathbf{x} \mid A\mathbf{x} = \mathbf{b}\,,\, a_i \leq x_i \leq b_i \ \forall i \neq j\,,\, d_{j+1} \leq x_j \leq b_j\right\} \tag{A.28}$$

respectively. The two new problems are now put into a collection of waiting nodes, or *pendant nodes.* We continue to work with the problem in one of these waiting nodes, which we call the present problem or *current problem.* When working with the current problem, one of the following situations will occur:

1. The current problem may be infeasible. In this case, no amount of further branching (*i.e.* further constraining) can bring the problem back to feasibility, and hence the problem is *fathomed as infeasible.*

2. The current problem might have an integral optimal solution $\hat{\mathbf{x}}$. If so, we compare $\mathbf{c}^T\hat{\mathbf{x}}$ with the best-so-far objective value $\bar{z}$ (which was initiated at ∞). If the new value is better (*i.e.* $\mathbf{c}^T\hat{\mathbf{x}} < \bar{z}$), we store $\hat{\mathbf{x}}$ as the best-so-far solution and update $\bar{z}$ so that $\bar{z} = \mathbf{c}^T\hat{\mathbf{x}}$. The

current problem is thus either *fathomed as currently optimal* or *fathomed as non-optimal*.

3. The current problem may have a non-integral solution $\hat{\mathbf{x}}$ with $\mathbf{c}^T\hat{\mathbf{x}} \geq \bar{z}$. In this case, the current problem cannot possibly contain an integral solution $\mathbf{x}$ with $\mathbf{c}^T\mathbf{x} < \bar{z}$, since the integer program is more constrained and cannot have a lower optimal objective value. The problem is thus *fathomed as bounded*.

4. The current problem may have a non-integral solution $\hat{\mathbf{x}}$ with $\mathbf{c}^T\hat{\mathbf{x}} < \bar{z}$. In this case, an optimal integral solution may exist, and hence we *branch*, creating two child nodes, which we add to the collection of pendant nodes.

As branching proceeds, the interval over which we solve the LP relaxation will eventually contain only one point. Therefore, sooner or later, we will come to the situation where either the problem is infeasible, or we are faced with an integral solution. Hence the algorithm will eventually stop (assuming the feasible space is bounded), yielding either the optimal integral solution or the information that no such solution exists.

Kall & Wallace [28] state that much research has been done on how to pick the correct variable x_j for branching, how to pick the branching value d_j, how to formulate the problem so that branching becomes simpler, and how to obtain a good initial integral solution so that $\bar{z} < \infty$.

A.4.2 Cutting Plane Methods

When we relax an integer program to obtain a linear program, we increase the solution space. However, all the points we add are non-integral. Cutting plane methods aim to add constraints to the relaxed LP to cut off some of these non-integral solutions, namely those that are not convex combinations of feasible integral points (and hence cannot contain any feasible integral points). These cuts are usually added in an iterative manner.

A cutting plane method runs through two major steps. The first step is to solve a relaxed LP, the second step is to evaluate the solution, and if it is not integral, to add cuts that cut away non-integral points, including the present solution. These cuts are then added to the relaxed LP, and the cycle is repeated. See Taha [43] for further details on cutting plane methods.

A.5 Software for Mathematical Programming

A wide range of commercial software is available for solving linear and nonlinear programs. Although the commercial software packages are very efficient and easy to use, they are usually restricted to solve problems up to a certain size, and their price depends on this size. This means that commercial software can be prohibitively expensive in solving deterministic equivalents of stochastic programs, since stochastic programs tend to become very large in size, particularly when there are *many* possible realizations of the random vector. Modelling languages are specially designed for formulating mathematical programs (linear programs, nonlinear programs and integer programs) in much the same way as they are written on paper. The modelling language then converts the formulation into a format that is accepted by a solver, which implements algorithms to solve the problem. Examples of such modelling languages are LINGO (see http://www.lindo.com), AMPL (see http://www.ampl.com/ampl), MPL (see http://www.maximal-usa.com), GAMS (see http://www.gams.com) and AIMMS (see http://www.paragon.nl). Well-known linear programming and integer programming solvers are CPLEX (see http://www.cplex.com), LINDO (see http://www.lindo.com) and XA (see http://www.sunsetsoft.com). Examples of nonlinear solvers are LINGO, CONOPT (see http://www.modeling.com), and LOQO (see http://www.princeton.edu/~rvdb). Spreadsheet solvers such as WHAT'S BEST (see http://www.lindo.com) and those by Frontline Systems (see http://www.frontsys.com) are becoming increasingly popular. Demo versions of some of these programs (such as LINGO, MPL and AIMMS) are available on the Internet. The motor racing example in Chapter 4, the stochastic formulation of the manufacturing design example in Chapter 4, and the airline planning example in Chapter 5 were solved using the student version of LINGO, which is capable of solving linear, nonlinear and integer programs with up to 100 constraints and 200 variables.

Public domain software for mathematical programming is far more limited. These programs generally tend to be less user friendly, and are often only available on Unix platforms. Most of these programs require input of the linear program in the form of MPS files (see ftp://plato.la.asu.edu/pub/mps_format.txt), a standard format that is accepted by most linear program solvers. One of the most powerful public domain programs for solving linear programs and mixed integer programs is LP_SOLVE (see ftp://ftp.es.ele.tue.nl/pub/lp_solve), which was used to solve the unit commitment problem in Chapter 6.

General purpose mathematical software can also be useful in mathematical programming. The Optimization Toolbox [36] in MATLAB (see http://www.mathworks.com) contains several procedures for optimization. The CONSTR procedure solves constrained nonlinear programs by sequential quadratic programming (see Section A.3.3) and was used to solve the water resource management problem in Chapter 2 and the production planning example in Chapter 3. The LP procedure was used in simulating the wait-and-see solutions in Chapter 3. All of the graphs in this dissertation were produced by MATLAB. Programs such as MAPLE (see http://www.maplesoft.com) and MATHEMATICA (see http://www.wolfram.com) are very good at solving simultaneous equations to an arbitrary degree of accuracy, and were also used to calculate various integrals. The MATHEMATICA function *FindMinimum* was used to solve the deterministic formulation of the manufacturing design problem in Chapter 4.

Not much software is available specially for stochastic programming. The IBM Optimization Subroutine Library (OSL - see http://www.research.ibm.com/osl) has a number of procedures for solving stochastic linear programs with recourse. These procedures must be linked with C or FORTRAN programs. AMPL has introduced a "scenario" command to aid the description of stochastic programs. Generally, stochastic programs can be solved by solving the deterministic equivalent problem with a standard linear or nonlinear program solver. This approach fails when the size of the deterministic equivalent problem is too large for the solver in question. As an alternative, you can always write your own algorithms!

Appendix B

Statistical Background

B.1 Probability Measure Theory

In order to derive some of the results in stochastic programming (particularly those concerning chance constraints - see Chapter 2), a basic knowledge of probability measure theory is required. This section provides an overview of probability measure theory based on Section 1.3.1 in Kall & Wallace [28].

In probability measure theory, we assume that we have a *sample space* Ω of *outcomes* ω (*e.g.* the results of random experiments), a collection $\mathcal{F}$ of subsets $F \subseteq \Omega$ called *events*, and a probability *measure* (or probability distribution) P assigning to each $F \in \mathcal{F}$ the probability $P(F)$ with which it occurs. It is required that

1. $\Omega \in \mathcal{F}$, and $F \in \mathcal{F} \Rightarrow \Omega - F \in \mathcal{F}$ (*i.e.* Ω is an event, and if F is an event, then so is its complement.)
2. If $F_i \in \mathcal{F}$, $i = 1, 2, \ldots$, then $\bigcup_{i=1}^{\infty} F_i \in \mathcal{F}$ (*i.e.* The countable union of events is an event.)
3. $0 \leq P(F) \leq 1 \; \forall F \in \mathcal{F}$ and $P(\Omega) = 1$ (*i.e.* The probability of any event lies between zero and one, and the sample space has a probability measure of one.)
4. If $F_i \in \mathcal{F}$, $i = 1, 2, \ldots$, and $F_i \cap F_j = \emptyset$ for $i \neq j$, then $P\left(\bigcup_{i=1}^{\infty} F_i\right) = \sum_{i=1}^{\infty} P(F_i)$ (*i.e.* the probability of a countable union of mutually exclusive events equals the sum of the probabilities of the events.)

Definition 24 (Probability Space) *The triple $(\Omega, \mathcal{F}, P)$ with the above properties is called a probability space.*

The concept of *natural measure* in measure theory is a generalization of the *length* of a set in one dimension, the *area* of a set in two dimensions and the *volume* of a set in three dimensions. A point (a zero-dimensional object) has a length of zero, a line (a one-dimensional object has an area of zero, and a plane (a two-dimensional object) has zero volume. Similarly, any set that can be adequately described in a certain number of dimensions has a zero natural measure in higher dimensions. Natural measure is really just a measure of the size (*i.e.* "hypervolume") of a set. We say that a set is *naturally measurable* if the "hypervolume" of the set is unique and can be measured by integrating unity (*i.e.* the constant 1) over the set, even if the result is infinite. An example of a set that is not naturally measurable is a set with *qualitative* elements rather than *quantitative* elements. We define $\mathcal{A}$ as the collection of all naturally measurable sets in $\mathbb{R}^k$.

In probability theory, we find random variables $\widetilde{\xi}$ and random vectors $\widetilde{\boldsymbol{\xi}}$.

Definition 25 (Random Vector) *A random vector is a vector-valued function*

$$\widetilde{\boldsymbol{\xi}} : \Omega \to \mathbb{R}^k \text{ such that } \forall A \in \mathcal{A}, \widetilde{\boldsymbol{\xi}}^{-1}(A) := \left\{\omega \mid \widetilde{\boldsymbol{\xi}}(\omega) \in A\right\} \in \mathcal{F} \tag{B.1}$$

This requires the "inverse" of any measurable set in $\mathbb{R}^k$ to be an event in Ω. Note that a random vector $\widetilde{\boldsymbol{\xi}} : \Omega \to \mathbb{R}^k$ induces a probability measure $P_{\widetilde{\boldsymbol{\xi}}}$ on $\mathcal{A}$ according to

$$P_{\widetilde{\boldsymbol{\xi}}}(A) = P\left(\left\{\omega \mid \widetilde{\boldsymbol{\xi}}(\omega) \in A\right\}\right) \; \forall A \in \mathcal{A} \tag{B.2}$$

Consider a random vector $\widetilde{\boldsymbol{\xi}}$ with the set $\Xi \in \mathcal{A}$ such that $\left\{\omega \mid \widetilde{\boldsymbol{\xi}}(\omega) \in \Xi\right\} = \Omega$. Note that $\Xi = \mathbb{R}^k$ always satisfies this, but there may be smaller sets in $\mathcal{A}$ that do so.

Definition 26 (Support) *The support Ξ of the probability measure P is the smallest closed set $\Xi \subseteq \mathbb{R}^k$ such that $P_{\widetilde{\boldsymbol{\xi}}}(\Xi) = 1$.*

With $\widehat{\mathcal{F}} = \{B \mid B = A \cap \Xi, \; A \in \mathcal{A}\}$, instead of the *abstract* probability space $(\Omega, \mathcal{F}, P)$ we may equivalently consider the *induced* probability space $\left(\Xi, \widehat{\mathcal{F}}, P\right)$ which we henceforth denote as

$(\Xi, \mathcal{F}, P)$.

Definition 27 (Almost Surely) *For a probability space $(\Xi, \mathcal{F}, P)$, if there is an event $N_\delta \in F$ with $P(N_\delta) = 0$ such that a property holds on $\Xi - N_\delta$, we say that the property holds almost surely (a.s.).*

Definition 28 (Expectation) *Consider a function $\psi : \Xi \to \mathbb{R}$. The expectation of $\psi(\widetilde{\xi})$ is given by the integral of the function $\psi(\boldsymbol{\xi})$ over the support Ξ of $\widetilde{\boldsymbol{\xi}}$ with respect to the probability measure P induced by the random vector $\widetilde{\boldsymbol{\xi}}$, i.e.*

$$E_{\widetilde{\xi}}\left[\psi(\widetilde{\xi})\right] = \int_{\Xi} \psi(\xi) dP(\xi)$$

In probability theory, the probability measure P of a probability space $(\Xi, \mathcal{F}, P)$ in $\mathbb{R}^k$ is equivalently described by the *distribution function*.

Definition 29 (Distribution Function) *The cumulative distribution function (c.d.f.) or distribution function $F_{\widetilde{\xi}}$ is defined by*

$$F_{\widetilde{\xi}}(\mathbf{x}) = P\left(\{\xi \mid \xi \leq \mathbf{x}\}\right), \mathbf{x} \in \mathbb{R}^k$$

Definition 30 (Density Function) *If there exists a function $f_{\widetilde{\xi}} : \Xi \to \mathbb{R}$ such that the distribution function can be represented as*

$$F_{\widetilde{\xi}}(\hat{\mathbf{x}}) = \int_{\mathbf{x} \leq \hat{\mathbf{x}}} f_{\widetilde{\xi}}(\mathbf{x}) d\mathbf{x}$$

then $f_{\widetilde{\xi}}$ is called the probability density function (p.d.f) of P and the distribution is called of continuous type.

B.2 Statistical Distributions

In this section, we give the probability density function (p.d.f.) and the cumulative distribution function (c.d.f.) for the various statistical distributions that were used in this dissertation. The mean and variance of each distribution are also given. The conditional expectations of the truncated versions of the uniform, exponential and normal distributions were used in the discretization procedure of Section 3.7.7.

B.2.1 Uniform Distributions

If $\tilde{\xi}$ has the *Uniform* distribution, $\tilde{\xi} \sim U(a,b)$ where $-\infty < a < b < \infty$, then $\tilde{\xi}$ has p.d.f.

$$f(\xi) = \left\{ \begin{array}{cl} \frac{1}{b-a} & \text{if } a \leq \xi \leq b \\ 0 & \text{otherwise} \end{array} \right] \tag{B.3}$$

and c.d.f.

$$F(\xi) = \left\{ \begin{array}{cl} 0 & \text{if } \xi \leq a \\ \frac{\xi-a}{b-a} & \text{if } a \leq \xi \leq b \\ 1 & \text{if } \xi \geq b \end{array} \right]$$

with mean

$$E\left(\tilde{\xi}\right) = \frac{a+b}{2}$$

and variance

$$\text{Var}\left(\tilde{\xi}\right) = \frac{(b-a)^2}{12}$$

The uniform distribution has support on the finite interval $[a,b]$. More generally, we can define the k-dimensional uniform distribution on a convex body $S \subset \mathbb{R}^k$ (with natural measure $\mu(S) > 0$), given by the density

$$f(\xi) = \left\{ \begin{array}{cl} \frac{1}{\mu(S)} & \text{if } \xi \in S \\ 0 & \text{if } \xi \notin S \end{array} \right] \tag{B.4}$$

Consider truncating the uniform distribution $U(a,b)$ on the left at A and on the right at B, where $a \leq A < B \leq b$. Then

$$\Pr\left[A \leq \tilde{\xi} \leq B \mid \tilde{\xi} \sim U(a,b)\right] = \frac{B-A}{b-a}$$

and hence

$$f(\xi \mid A \leq \xi \leq B) \;=\; \left\{ \begin{array}{cl} \frac{1}{B-A} & \text{if } A \leq \xi \leq B \\ 0 & \text{otherwise} \end{array} \right]$$

$$F(\xi \mid A \leq \xi \leq B) = \left\{ \begin{array}{cc} 0 & \text{if } \xi \leq A \\ \frac{\xi - A}{B - A} & \text{if } A \leq \xi \leq B \\ 1 & \text{if } \xi \geq B \end{array} \right]$$

$$E\left(\tilde{\xi} \mid A \leq \xi \leq B\right) = \frac{A+B}{2}$$

In other words, $\left(\tilde{\xi} \mid A \leq \tilde{\xi} \leq B, \tilde{\xi} \sim U(a,b)\right) \sim U(A,B)$.

B.2.2 Exponential Distributions

If $\tilde{\xi}$ has the *Exponential* distribution, $\tilde{\xi} \sim Exp(\lambda)$ where $\lambda > 0$, then $\tilde{\xi}$ has p.d.f.

$$f(\xi) = \left\{ \begin{array}{cc} 0 & \text{if } \xi < 0 \\ \lambda e^{-\lambda \xi} & \text{if } \xi \geq 0 \end{array} \right] \quad \text{(B.5)}$$

and c.d.f.

$$F(\xi) = \left\{ \begin{array}{cc} 0 & \text{if } \xi \leq 0 \\ 1 - e^{-\lambda \xi} & \text{if } \xi \geq 0 \end{array} \right]$$

with mean

$$E\left(\tilde{\xi}\right) = \frac{1}{\lambda}$$

and variance

$$\text{Var}\left(\tilde{\xi}\right) = \frac{1}{\lambda^2}$$

The exponential distribution has non-negative support. Consider truncating the exponential distribution $Exp(\lambda)$ on the left at A and on the right at B where $0 \leq A < B < \infty$. Then

$$\Pr\left[A \leq \tilde{\xi} \leq B \mid \tilde{\xi} \sim Exp(\lambda)\right] = e^{-\lambda A} - e^{-\lambda B}$$

and hence

$$f(\xi \mid A \leq \xi \leq B) = \left\{ \begin{array}{cc} \frac{\lambda e^{-\lambda \xi}}{e^{-\lambda A} - e^{-\lambda B}} & \text{if } A \leq \xi \leq B \\ 0 & \text{otherwise} \end{array} \right]$$

$$F(\xi \mid A \leq \xi \leq B) = \begin{cases} 0 & \text{if } \xi \leq A \\ \frac{e^{-\lambda A} - e^{-\lambda \xi}}{e^{-\lambda A} - e^{-\lambda B}} & \text{if } A \leq \xi \leq B \\ 1 & \text{if } \xi \geq B \end{cases}$$

$$E\left(\widetilde{\xi} \mid A \leq \xi \leq B\right) = \frac{(1 + \lambda A)e^{-\lambda A} - (1 + \lambda B)e^{-\lambda B}}{(e^{-\lambda A} - e^{-\lambda B})\lambda} \tag{B.6}$$

B.2.3 Normal Distributions

If $\widetilde{\xi}$ has the *Normal* distribution, $\widetilde{\xi} \sim N(\mu, \sigma^2)$ where $-\infty < \mu < \infty$ and $\sigma^2 > 0$, then $\widetilde{\xi}$ has p.d.f.

$$f(\xi) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{1}{2}\left(\frac{\xi - \mu}{\sigma}\right)^2\right] = \frac{1}{\sigma}\phi\left(\frac{\xi - \mu}{\sigma}\right), \ -\infty < \xi < \infty \tag{B.7}$$

where

$$\phi(z) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}z^2} \tag{B.8}$$

and c.d.f.

$$F(\xi) = \Phi\left(\frac{\xi - \mu}{\sigma}\right), \ -\infty < \xi < \infty$$

where

$$\Phi(z) = \int_{-\infty}^{z} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}\xi^2} d\xi = \int_{-\infty}^{z} \phi(\xi) d\xi \tag{B.9}$$

The Normal distribution has support on the real line, and has mean $E(\widetilde{\xi}) = \mu$ and variance $\text{Var}(\widetilde{\xi}) = \sigma^2$.

More generally, if $\widetilde{\boldsymbol{\xi}}$ has a *Multivariate Normal* distribution $\widetilde{\boldsymbol{\xi}} \sim N_k(\boldsymbol{\mu}, \Sigma)$, where $\boldsymbol{\mu} \in \mathbb{R}^k$ and $\Sigma > O$ (*i.e.* Σ is a positive definite[1] $k \times k$ matrix), then $\widetilde{\boldsymbol{\xi}}$ has the p.d.f.

$$f(\boldsymbol{\xi}) = (2\pi)^{-\frac{k}{2}} |\Sigma|^{-\frac{1}{2}} \exp\left[-\frac{1}{2}(\boldsymbol{\xi} - \boldsymbol{\mu})^T \Sigma^{-1} (\boldsymbol{\xi} - \boldsymbol{\mu})\right], \ \boldsymbol{\xi} \in \mathbb{R}^k \tag{B.10}$$

where $|\Sigma| = \det(\Sigma)$. The distribution has mean $E(\widetilde{\boldsymbol{\xi}}) = \boldsymbol{\mu}$ and variance-covariance matrix $\text{Cov}(\widetilde{\boldsymbol{\xi}}) = \Sigma$. It is well known that a linear transformation of a multivariate normal distribution

[1]A real symmetric matrix Σ is said to be *positive definite* (denoted by $\Sigma > O$) if $\mathbf{x}^T \Sigma \mathbf{x} > 0$ for every $\mathbf{x} \neq \mathbf{0}$.

also has a multivariate normal distribution, *i.e.*

$$\tilde{\xi} \sim N_k(\mu, \Sigma) \Rightarrow A\tilde{\xi} \sim N_m\left(A\mu, A\Sigma A^T\right) \tag{B.11}$$

where A is an $m \times k$ matrix.

Consider *truncating* the normal distribution $N(\mu, \sigma^2)$ on the left at A and on the right at B here $-\infty < A < B < \infty$. Then

$$\Pr\left[A \leq \tilde{\xi} \leq B \mid \tilde{\xi} \sim N(\mu, \sigma^2)\right] = \Phi\left(\frac{B-\mu}{\sigma}\right) - \Phi\left(\frac{A-\mu}{\sigma}\right)$$

and hence

$$\begin{aligned}
f(\xi \mid A \leq \xi \leq B) &= \left\{ \begin{array}{cc} \frac{\frac{1}{\sqrt{2\pi}\sigma}\exp\left[-\frac{1}{2}\left(\frac{\xi-\mu}{\sigma}\right)^2\right]}{\Phi\left(\frac{B-\mu}{\sigma}\right)-\Phi\left(\frac{A-\mu}{\sigma}\right)} & \text{if } A \leq \xi \leq B \\ 0 & \text{otherwise} \end{array} \right] \\
F(\xi \mid A \leq \xi \leq B) &= \left\{ \begin{array}{cc} 0 & \text{if } -\infty < \xi \leq A \\ \frac{\Phi\left(\frac{\xi-\mu}{\sigma}\right)-\Phi\left(\frac{A-\mu}{\sigma}\right)}{\Phi\left(\frac{B-\mu}{\sigma}\right)-\Phi\left(\frac{A-\mu}{\sigma}\right)} & \text{if } A \leq \xi \leq B \\ 1 & \text{if } B \leq \xi < \infty \end{array} \right] \\
E\left(\tilde{\xi} \mid A \leq \xi \leq B\right) &= \mu + \sigma\frac{\left[\phi\left(\frac{A-\mu}{\sigma}\right) - \phi\left(\frac{B-\mu}{\sigma}\right)\right]}{\left[\Phi\left(\frac{B-\mu}{\sigma}\right) - \Phi\left(\frac{A-\mu}{\sigma}\right)\right]}
\end{aligned} \tag{B.12}$$

B.2.4 Gamma Distribution

If $\tilde{\xi}$ has the *Gamma* distribution $\tilde{\xi} \sim G(r, \lambda)$ where $r > 0$ and $\lambda > 0$, then $\tilde{\xi}$ has p.d.f.

$$f(\xi) = \left\{ \begin{array}{cc} \frac{\lambda^r}{\Gamma(r)}\xi^{r-1}e^{-\lambda\xi} & \text{if } \xi > 0 \\ 0 & \text{if } \xi \leq 0 \end{array} \right] \tag{B.13}$$

where $\Gamma(\cdot)$ is the *gamma function* defined as $\Gamma(r) = \int_0^\infty t^{r-1}e^{-t}dt$. The gamma distribution has positive support and has mean

$$E\left(\tilde{\xi}\right) = \frac{r}{\lambda}$$

and variance

$$\operatorname{Var}\left(\tilde{\xi}\right) = \frac{r}{\lambda^2}$$

B.2.5 Lognormal Distribution

If $\tilde{\xi}$ has the *Lognormal* distribution $\tilde{\xi} \sim LN(\mu, \sigma^2)$ where $-\infty < \mu < \infty$ and $\sigma^2 > 0$, then $\tilde{\xi}$ has p.d.f.

$$f(\xi) = \left\{ \begin{array}{cc} \frac{1}{\xi\sqrt{2\pi}\sigma} \exp\left[-\frac{1}{2\sigma^2}(\ln \xi - \mu)^2\right] & \text{if } \xi > 0 \\ 0 & \text{if } \xi \le 0 \end{array} \right] \tag{B.14}$$

and c.d.f.

$$F(\xi) = \left\{ \begin{array}{cc} \Phi\left(\frac{\ln \xi - \mu}{\sigma}\right) & \text{if } \xi > 0 \\ 0 & \text{if } \xi \le 0 \end{array} \right]$$

where $\Phi(\cdot)$ is given by (B.9). The lognormal distribution has positive support and has mean

$$E\left(\tilde{\xi}\right) = e^{\mu + \frac{\sigma^2}{2}}$$

and variance

$$\operatorname{Var}\left(\tilde{\xi}\right) = e^{2\mu+\sigma^2}\left(e^{\sigma^2} - 1\right)$$

B.2.6 Weibull Distribution

If $\tilde{\xi}$ has the *Weibull* distribution $\tilde{\xi} \sim W(\alpha, \beta)$, where $\alpha > 0$ and $\beta > 0$, then $\tilde{\xi}$ has p.d.f.

$$f(\xi) = \left\{ \begin{array}{cc} \alpha\beta^{-\alpha}\xi^{\alpha-1} \exp\left[-\left(\frac{\xi}{\beta}\right)^{\alpha}\right] & \text{if } \xi > 0 \\ 0 & \text{if } \xi \le 0 \end{array} \right] \tag{B.15}$$

and c.d.f.

$$F(\xi) = \left\{ \begin{array}{cc} 1 - \exp\left[-\left(\frac{\xi}{\beta}\right)^{\alpha}\right] & \text{if } \xi > 0 \\ 0 & \text{if } \xi \le 0 \end{array} \right]$$

The Weibull distribution has positive support and has mean

$$E\left(\tilde{\xi}\right) = \beta\Gamma\left(1 + \frac{1}{\alpha}\right)$$

and variance

$$\mathrm{Var}\left(\tilde{\xi}\right) = \beta^2 \left[\Gamma\left(1+\frac{2}{\alpha}\right) - \left(\Gamma\left(1+\frac{1}{\alpha}\right)\right)^2\right]$$

B.2.7 Point Distribution

If $\tilde{\xi}$ has the *Point* distribution $\tilde{\xi} \sim Pt(c)$, where $-\infty < c < \infty$, then $\tilde{\xi}$ has the probability *mass* function (p.m.f.)

$$f(\xi) = \left\{ \begin{array}{ll} 1 & \text{if } \xi = c \\ 0 & \text{if } \xi \neq c \end{array} \right]$$

and c.d.f.

$$F(\xi) = \left\{ \begin{array}{ll} 0 & \text{if } \xi < c \\ 1 & \text{if } \xi \geq c \end{array} \right]$$

The distribution has mean $E(\tilde{\xi}) = c$ and variance $\mathrm{Var}(\tilde{\xi}) = 0$. The Point distribution is a degenerate distribution and has support at one discrete point only. It is essentially a "deterministic" distribution.

B.2.8 Poisson Distribution

If $\tilde{\xi}$ has the *Poisson* distribution $\tilde{\xi} \sim P(\lambda)$, where $\lambda > 0$, then $\tilde{\xi}$ has p.m.f.

$$f(\xi) = \left\{ \begin{array}{ll} e^{-\lambda}\frac{\lambda^{\xi}}{\xi!} & \text{if } \xi = 0, 1, \ldots \\ 0 & \text{elsewhere} \end{array} \right] \tag{B.16}$$

and c.d.f.

$$F(\xi) = \left\{ \begin{array}{ll} 0 & \text{if } \xi < 0 \\ e^{-\lambda} \sum_{i=0}^{\lfloor \xi \rfloor} \frac{\lambda^i}{i!} & \text{if } \xi \geq 0 \end{array} \right]$$

The distribution has mean $E(\tilde{\xi}) = \lambda$ and variance $\mathrm{Var}(\tilde{\xi}) = \lambda$. The Poisson distribution is a discrete distribution with support on the non-negative integers.

B.3 Standard Integrals

The following integrals were obtained with the aid of MATHEMATICA and were used in the calculations of Section 3.7.5.

$$\int \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right] dx = \Phi\left(\frac{x-\mu}{\sigma}\right) + k$$

$$\int \Phi(ax+b)dx = \frac{1}{\sqrt{2\pi}a} \exp\left[-\frac{1}{2}(ax+b)^2\right] + \left(x + \frac{b}{a}\right)\Phi(ax+b) + k$$

$$\int x\Phi(ax+b)dx = \frac{ax-b}{2a^2\sqrt{2\pi}} \exp\left[-\frac{1}{2}(ax+b)^2\right] + \frac{1}{2}\left(x^2 - \frac{b^2+1}{a^2}\right)\Phi(ax+b) + k$$

$$\int e^{-cx}\Phi(ax+b)dx = -\frac{1}{c}e^{-cx}\Phi(ax+b) + \frac{1}{c}\exp\left[\frac{c}{2a^2}(2ab+c)\right]\Phi\left(ax+b+\frac{c}{a}\right) + k$$

$$\int e^{-ax^2+bx}dx = \sqrt{\frac{\pi}{a}} \exp\left[\frac{b^2}{4a}\right]\Phi\left(\frac{2ax-b}{\sqrt{2a}}\right) + k$$

$$\int xe^{-cx}dx = -\frac{1}{c^2}e^{-cx} - \frac{x}{c}e^{-cx} + k$$

$$\begin{aligned}\int xe^{-cx}\Phi(ax+b)dx &= -\frac{1}{ac\sqrt{2\pi}} \exp\left[-\frac{1}{2}b^2 - (ab+c)x - \frac{1}{2}a^2x^2\right] - \frac{1+cx}{c^2}e^{-cx}\Phi(ax+b) \\ &\quad + \left(\frac{1}{c^2} - \frac{b}{ac} - \frac{1}{a^2}\right)\exp\left[\frac{c}{2a^2}(2ab+c)\right]\Phi\left(ax+b+\frac{c}{a}\right) + k\end{aligned}$$

Appendix C

Solutions to the Unit Commitment Problem

Appendix C.1

Solution to EV Problem

i	1	2	3	4	5	6	7	8	9	10	11	12	13	
Unit	Arnot	Duvha	Hendrin	Kendal	Kriel	Lethabo	Matimba	Matla	Tutuka	Majuba	Koeberg	Gas	Unmet	Sum
u'(i)	2	4	7	4	4	4	6	6	6	0	2	0	0	
u(i)	2	4	8	4	4	0	6	6	6	0	2	0	0	
v(i)	0	0	1	0	0	0	0	0	0	0	0	0	0	
g(i)u(i)	400	1400	1040	1280	1020	0	2160	1950	1650	0	1840	0	0	Capacity
G(i)u(i)	660	2300	1520	2560	1900	0	3690	3450	3510	0	1840	0	∞	21430
t	*x(1,t)*	*x(2,t)*	*x(3,t)*	*x(4,t)*	*x(5,t)*	*x(6,t)*	*x(7,t)*	*x(8,t)*	*x(9,t)*	*x(10,t)*	*x(11,t)*	*x(12,t)*	*x(13,t)*	*y(t)*
1	400	2300	1040	1280	1020	0	3690	1950	2007	0	1840	0	0	5903
2	400	2300	1040	1280	1020	0	3690	1950	1866	0	1840	0	0	6044
3	400	2300	1040	1280	1020	0	3690	1950	1730	0	1840	0	0	6180
4	400	2300	1040	1280	1020	0	3690	1950	1828	0	1840	0	0	6082
5	400	2300	1040	1280	1020	0	3690	1950	2133	0	1840	0	0	5777
6	400	2300	1040	1280	1020	0	3690	1950	3362	0	1840	0	0	4548
7	660	2300	1520	1462	1900	0	3690	1950	3510	0	1840	0	0	2598
8	660	2300	1520	2560	1900	0	3690	2444	3510	0	1840	0	0	1006
9	660	2300	1520	2560	1900	0	3590	3385	3510	0	1840	0	0	65
10	660	2300	1520	2560	1900	0	3690	3385	3510	0	1840	0	0	65
11	660	2300	1520	2560	1900	0	3690	3113	3510	0	1840	0	0	337
12	660	2300	1520	2560	1900	0	3690	3177	3510	0	1840	0	0	273
13	660	2300	1520	2560	1900	0	3690	2987	3510	0	1840	0	0	463
14	660	2300	1520	2560	1900	0	3690	2731	3510	0	1840	0	0	719
15	660	2300	1520	2560	1900	0	3690	2602	3510	0	1840	0	0	848
16	660	2300	1520	2560	1900	0	3690	2806	3510	0	1840	0	0	644
17	660	2300	1520	2560	1900	0	3690	2615	3510	0	1840	0	0	835
18	660	2300	1520	2560	1900	0	3690	2133	3510	0	1840	0	0	1317
19	660	2300	1520	2560	1900	0	3690	2225	3510	0	1840	0	0	1225
20	660	2300	1520	2560	1900	0	3690	3116	3510	0	1840	0	0	334
21	660	2300	1520	2560	1900	0	3690	3301	3510	0	1840	0	0	149
22	660	2300	1520	2560	1900	0	3690	2175	3510	0	1840	0	0	1275
23	660	2300	1520	1386	1900	0	3690	1950	3510	0	1840	0	0	2674
24	400	2300	1520	1280	1218	0	3690	1950	3510	0	1840	0	0	3722
Switch-on cost	0	0	17500	0	0	0	0	0	0	0	0	0	0	17500
Fixed cost	12979	25693	21744	45510	38352	0	25906	61242	66030	0	0	0	0	297456
Marginal cost	103064	279373	225142	372692	291148	0	232899	502255	507350	0	618240	0	0	3132162
Total unit cost	116043	305066	264386	418202	329500	0	258804	563497	573380	0	618240	0	0	3447117

i	1	2	3	4	5	6	7	8	9	10	11	12	13	
Unit	Arnot	Duvha	Hendrina	Kendal	Kriel	Lethabo	Matimba	Matla	Tutuka	Majuba	Koeberg	Gas	Unmet	Sum
u'(i)	2	4	7	4	4	4	6	6	6	0	2	0	0	
u(i)	2	4	7	4	4	2	6	6	6	0	2	1	0	
v(i)	0	0	0	0	0	0	0	0	0	0	0	1	0	
g(i)u(i)	400	1400	910	1280	1020	750	2160	1950	1650	0	1840	0	0	Capacity
G(i)u(i)	660	2300	1330	2560	1900	1186	3690	3450	3510	0	1840	57	∞	22426
Switch-on cost	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Fixed cost	12979	25693	19026	45510	38352	13962	25906	61242	66030	0	0	0	0	308699
Scenario 1	0.054489													
t	*x(1,t,1)*	*x(2,t,1)*	*x(3,t,1)*	*x(4,t,1)*	*x(5,t,1)*	*x(6,t,1)*	*x(7,t,1)*	*x(8,t,1)*	*x(9,t,1)*	*x(10,t,1)*	*x(11,t,1)*	*x(12,t,1)*	*x(13,t,1)*	*y(t,1)*
1	400	1400	910	1280	1020	750	3551	1950	1650	0	1840	0	0	7675
2	400	1400	910	1280	1020	750	3417	1950	1650	0	1840	0	0	7809
3	400	1400	910	1280	1020	750	3288	1950	1650	0	1840	0	0	7939
4	400	1400	910	1280	1020	750	3381	1950	1650	0	1840	0	0	7845
5	400	1400	910	1280	1020	750	3670	1950	1650	0	1840	0	0	7556
6	400	2300	910	1280	1020	750	3690	1950	1898	0	1840	0	0	6388
7	400	2300	1150	1280	1020	750	3690	1950	3510	0	1840	0	0	4536
8	660	2300	1330	1473	1900	750	3690	1950	3510	0	1840	0	0	3023
9	660	2300	1330	2367	1900	750	3690	1950	3510	0	1840	0	0	2129
10	660	2300	1330	2367	1900	750	3690	1950	3510	0	1840	0	0	2129
11	660	2300	1330	2108	1900	750	3690	1950	3510	0	1840	0	0	2388
12	660	2300	1330	2169	1900	750	3690	1950	3510	0	1840	0	0	2327
13	660	2300	1330	1989	1900	750	3690	1950	3510	0	1840	0	0	2507
14	660	2300	1330	1746	1900	750	3690	1950	3510	0	1840	0	0	2751
15	660	2300	1330	1623	1900	750	3690	1950	3510	0	1840	0	0	2873
16	660	2300	1330	1817	1900	750	3690	1950	3510	0	1840	0	0	2679
17	660	2300	1330	1635	1900	750	3690	1950	3510	0	1840	0	0	2861
18	557	2300	1330	1280	1900	750	3690	1950	3510	0	1840	0	0	3319
19	645	2300	1330	1280	1900	750	3690	1950	3510	0	1840	0	0	3231
20	660	2300	1330	2111	1900	750	3690	1950	3510	0	1840	0	0	2385
21	660	2300	1330	2287	1900	750	3690	1950	3510	0	1840	0	0	2209
22	597	2300	1330	1280	1900	750	3690	1950	3510	0	1840	0	0	3279
23	400	2300	1078	1280	1020	750	3690	1950	3510	0	1840	0	0	4608
24	400	2300	910	1280	1020	750	3690	1950	2683	0	1840	0	0	5603
Unit supply	13319	50700	28549	39051	37680	18000	87416	46800	72501	0	44160	0	0	438175
Marginal cost	97914	256598	191294	289875	276786	168810	2298[illegible]	393431	483314	0	618240	0	0	3006132
Scenario 2	0.244201													
t	*x(1,t,2)*	*x(2,t,2)*	*x(3,t,2)*	*x(4,t,2)*	*x(5,t,2)*	*x(6,t,2)*	*x(7,t,2)*	*x(8,t,2)*	*x(9,t,2)*	*x(10,t,2)*	*x(11,t,2)*	*x(12,t,2)*	*x(13,t,2)*	*y(t,2)*

1	400	1649	910	1280	1020	750	3690	1950	1650	0	1840	0	0	7287
2	400	1511	910	1280	1020	750	3690	1950	1650	0	1840	0	0	7425
3	400	1400	910	1280	1020	750	3669	1950	1650	0	1840	0	0	7557
4	400	1474	910	1280	1020	750	3690	1950	1650	0	1840	0	0	7462
5	400	1772	910	1280	1020	750	3690	1950	1650	0	1840	0	0	7164
6	400	2300	910	1280	1020	750	3690	1950	2320	0	1840	0	0	5966
7	400	2300	1330	1280	1311	750	3690	1950	3510	0	1840	0	0	4065
8	660	2300	1330	1983	1900	750	3690	1950	3510	0	1840	0	0	2513
9	660	2300	1330	2560	1900	750	3690	2291	3510	0	1840	0	0	1595
10	660	2300	1330	2560	1900	750	3690	2291	3510	0	1840	0	0	1595
11	660	2300	1330	2560	1900	750	3690	2026	3510	0	1840	0	0	1860
12	660	2300	1330	2560	1900	750	3690	2088	3510	0	1840	0	0	1798
13	660	2300	1330	2513	1900	750	3690	1950	3510	0	1840	0	0	1983
14	660	2300	1330	2263	1900	750	3690	1950	3510	0	1840	0	0	2233
15	660	2300	1330	2138	1900	750	3690	1950	3510	0	1840	0	0	2359
16	660	2300	1330	2336	1900	750	3690	1950	3510	0	1840	0	0	2160
17	660	2300	1330	2150	1900	750	3690	1950	3510	0	1840	0	0	2346
18	660	2300	1330	1680	1900	750	3690	1950	3510	0	1840	0	0	2816
19	660	2300	1330	1770	1900	750	3690	1950	3510	0	1840	0	0	2726
20	660	2300	1330	2560	1900	750	3690	2029	3510	0	1840	0	0	1857
21	660	2300	1330	2560	1900	750	3690	2209	3510	0	1840	0	0	1677
22	660	2300	1330	1721	1900	750	3690	1950	3510	0	1840	0	0	2775
23	400	2300	1330	1280	1237	750	3690	1950	3510	0	1840	0	0	4139
24	400	2300	910	1280	1020	750	3690	1950	3125	0	1840	0	0	5161
Unit supply	13500	51506	28980	45435	38188	18000	88539	48033	73365	0	44160	0	0	449706
Marginal cost	99242	260678	194185	337258	280500	168810	232843	403798	489080	0	618240	0	0	3084633

Scenario 3	0.40262													
t	*x(1,t,3)*	*x(2,t,3)*	*x(3,t,3)*	*x(4,t,3)*	*x(5,t,3)*	*x(6,t,3)*	*x(7,t,3)*	*x(8,t,3)*	*x(9,t,3)*	*x(10,t,3)*	*x(11,t,3)*	*x(12,t,3)*	*x(13,t,3)*	*y(t,3)*
1	400	2037	910	1280	1020	750	3690	1950	1650	0	1840	0	0	6899
2	400	1896	910	1280	1020	750	3690	1950	1650	0	1840	0	0	7040
3	400	1760	910	1280	1020	750	3690	1950	1650	0	1840	0	0	7176
4	400	1858	910	1280	1020	750	3690	1950	1650	0	1840	0	0	7078
5	400	2163	910	1280	1020	750	3690	1950	1650	0	1840	0	0	6773
6	400	2300	910	1280	1020	750	3690	1950	2742	0	1840	0	0	5544
7	400	2300	1330	1280	1782	750	3690	1950	3510	0	1840	0	0	3594
8	660	2300	1330	2494	1900	750	3690	1950	3510	0	1840	0	0	2002
9	660	2300	1330	2560	1900	750	3690	2825	3510	0	1840	0	0	1061
10	660	2300	1330	2560	1900	750	3690	2825	3510	0	1840	0	0	1061
11	660	2300	1330	2560	1900	750	3690	2553	3510	0	1840	0	0	1333
12	660	2300	1330	2560	1900	750	3690	2617	3510	0	1840	0	0	1269

13	660	2300	1330	2560	1900	750	3690	2427	3510	0	1840	0	0	1459
14	660	2300	1330	2560	1900	750	3690	2171	3510	0	1840	0	0	1715
15	660	2300	1330	2560	1900	750	3690	2042	3510	0	1840	0	0	1844
16	660	2300	1330	2560	1900	750	3690	2246	3510	0	1840	0	0	1640
17	660	2300	1330	2560	1900	750	3690	2055	3510	0	1840	0	0	1831
18	560	2300	1330	2183	1900	750	3690	1950	3510	0	1840	0	0	2313
19	660	2300	1330	2275	1900	750	3690	1950	3510	0	1840	0	0	2221
20	660	2300	1330	2560	1900	750	3690	2556	3510	0	1840	0	0	1330
21	660	2300	1330	2560	1900	750	3690	2741	3510	0	1840	0	0	1145
22	660	2300	1330	2225	1900	750	3690	1950	3510	0	1840	0	0	2271
23	400	2300	1330	1280	1706	750	3690	1950	3510	0	1840	0	0	3670
24	400	2300	968	1280	1020	750	3690	1950	3510	0	1840	0	0	4718
Unit supply	13500	53414	29038	48857	39128	18000	88560	52408	74172	0	44160	0	0	461237
Marginal cost	99242	270334	194573	362664	287402	168810	232899	440575	494457	0	618240	0	3	3169196

Scenario 4	0.244201													
t	*x(1,t,4)*	*x(2,t,4)*	*x(3,t,4)*	*x(4,t,4)*	*x(5,t,4)*	*x(6,t,4)*	*x(7,t,4)*	*x(8,t,4)*	*x(9,t,4)*	*x(10,t,4)*	*x(11,t,4)*	*x(12,t,4)*	*x(13,t,4)*	*y(t,4)*
1	400	2300	910	1280	1020	750	3690	1950	1775	0	1840	0	0	6511
2	400	2281	910	1280	1020	750	3690	1950	1650	0	1840	0	0	6655
3	400	2142	910	1280	1020	750	3690	1950	1650	0	1840	0	0	6794
4	400	2242	910	1280	1020	750	3690	1950	1650	0	1840	0	0	6694
5	400	2300	910	1280	1020	750	3690	1950	1904	0	1840	0	0	6382
6	400	2300	910	1280	1020	750	3690	1950	3164	0	1840	0	0	5122
7	660	2300	1330	1373	1900	750	3690	1950	3510	0	1840	0	0	3123
8	660	2300	1330	2560	1900	750	3690	2395	3510	0	1840	0	0	1491
9	660	2300	1330	2560	1900	750	3690	3359	3510	0	1840	0	0	527
10	660	2300	1330	2560	1900	750	3690	3359	3510	0	1840	0	0	527
11	660	2300	1330	2560	1900	750	3690	3080	3510	0	1840	0	0	806
12	660	2300	1330	2560	1900	750	3690	3146	3510	0	1840	0	0	740
13	660	2300	1330	2560	1900	750	3690	2951	3510	0	1840	0	0	935
14	660	2300	1330	2560	1900	750	3690	2689	3510	0	1840	0	0	1197
15	660	2300	1330	2560	1900	750	3690	2557	3510	0	1840	0	0	1330
16	660	2300	1330	2560	1900	750	3690	2766	3510	0	1840	0	0	1120
17	660	2300	1330	2560	1900	750	3690	2570	3510	0	1840	0	0	1316
18	660	2300	1330	2560	1900	750	3690	2076	3510	0	1840	0	0	1810
19	660	2300	1330	2560	1900	750	3690	2170	3510	0	1840	0	0	1716
20	660	2300	1330	2560	1900	750	3690	3083	3510	0	1840	0	0	803
21	660	2300	1330	2560	1900	750	3690	3273	3510	0	1840	0	0	613
22	660	2300	1330	2560	1900	750	3690	2119	3510	0	1840	0	0	1767
23	680	2300	1330	1295	1900	750	3690	1950	3510	0	1840	0	0	3201
24	400	2300	1330	1280	1101	750	3690	1950	3510	0	1840	0	0	4275

Unit supply	14020	54965	29400	50028	39521	18000	88560	59142	74974	0	44160	0	0	472769
Marginal cost	103064	278181	196999	371354	290286	168810	232899	497188	499800	0	618240	0	0	3256822
Scenario 5	0.054489													
i	*x(1,i,5)*	*x(2,i,5)*	*x(3,i,5)*	*x(4,i,5)*	*x(5,i,5)*	*x(6,i,5)*	*x(7,i,5)*	*x(8,i,5)*	*x(9,i,5)*	*x(10,i,5)*	*x(11,i,5)*	*x(12,i,5)*	*x(13,i,5)*	*y(i,5)*
1	400	2300	910	1280	1020	750	3690	195?	2163	0	1840	0	0	6123
2	400	2300	910	1280	1020	750	3690	1950	2015	0	1840	0	0	6271
3	400	2300	910	1280	1020	750	3690	1950	1873	0	1840	0	0	6414
4	400	2300	910	1280	1020	750	3690	1950	1975	0	1840	0	0	6311
5	400	2300	910	1280	1020	750	3690	1950	2296	0	1840	0	0	5990
6	400	2300	986	1280	1020	750	3690	1950	3510	0	1840	0	0	4700
7	660	2300	1330	1844	1900	750	3690	1950	3510	0	1840	0	0	2052
8	660	2300	1330	2560	1900	750	3690	2905	3510	0	1840	0	0	981
9	660	2300	1330	2560	1900	1186	3690	3450	3510	0	1840	7	0	-7
10	660	2300	1330	2560	1900	1186	3690	3450	3510	0	1840	7	0	-7
11	660	2300	1330	2560	1900	908	3690	3450	3510	0	1840	0	0	278
12	660	2300	1330	2560	1900	975	3690	3450	3510	0	1840	0	0	211
13	660	2300	1330	2560	1900	775	3690	3450	3510	0	1840	0	0	411
14	660	2300	1330	2560	1900	750	3690	3207	3510	0	1840	0	0	680
15	660	2300	1330	2560	1900	750	3690	3071	3510	0	1840	0	0	815
16	660	2300	1330	2560	1900	750	3690	3285	3510	0	1840	0	0	601
17	660	2300	1330	2560	1900	750	3690	3085	3510	0	1840	0	0	801
18	660	2300	1330	2560	1900	750	3690	2579	3510	0	1840	0	0	1307
19	660	2300	1330	2560	1900	750	3690	2675	3510	0	1840	0	0	1211
20	660	2300	1330	2560	1900	911	3690	3450	3510	0	1840	0	0	275
21	660	2300	1330	2560	1900	1105	3690	3450	3510	0	1840	0	0	81
22	660	2300	1330	2560	1900	750	3690	2623	3510	0	1840	0	0	1263
23	660	2300	1330	1764	1900	750	3690	1950	3510	0	1840	0	0	2732
24	400	2300	1330	1280	1543	750	3690	1950	3510	0	1840	0	0	3633
Unit supply	14020	55200	29476	50967	39963	19796	88530	65130	77012	0	44160	15	0	484299
Marginal cost	103064	279373	197509	378329	293538	185651	232899	547521	513392	0	618240	7141	0	3366656
Expected marginal	100311	269636	195052	355469	286175	169728	232721	448678	494873	0	618240	389	0	3171273
Total cost	113290	295329	214078	400979	324527	183690	258627	509920	560903	0	618240	389	0	3479972

Expected Value of EV Solution (EEV)

i	1	2	3	4	5	6	7	8	9	10	11	12	13	
Unit	Arnot	Duvha	Hendrina	Kendal	Kriel	Lethabo	Matimba	Matla	Tutuka	Majuba	Koeberg	Gas	Unmet	Sum
u'(i)	2	4	7	4	4	4	6	6	6	0	2	0	0	
u(i)	2	4	8	4	4	0	6	6	6	0	2	6	1	
v(i)	0	0	1	0	0	0	0	0	0	0	0	6	1	
g(i)u(i)	400	1400	1040	1280	1020	0	2160	1950	1650	0	1840	0	0	Capacity
G(i)u(i)	660	2300	1520	2560	1900	0	3690	3450	3510	0	1840	342	∞	21430
Switch-on cost	0	0	17500	0	0	0	0	0	0	0	0	0	0	17500
Fixed cost	12979	25693	21744	45510	38352	0	25906	61242	66030	0	0	0	0	297455
Scenario 1	0.054489													
t	*x(1,t,1)*	*x(2,t,1)*	*x(3,t,1)*	*x(4,t,1)*	*x(5,t,1)*	*x(6,t,1)*	*x(7,t,1)*	*x(8,t,1)*	*x(9,t,1)*	*x(10,t,1)*	*x(11,t,1)*	*x(12,t,1)*	*x(13,t,1)*	*y(t,1)*
1	400	1881	1040	1280	1020	0	3690	1950	1650	0	1840	0	0	6679
2	400	1747	1040	1280	1020	0	3690	1950	1650	0	1840	0	0	6813
3	400	1618	1040	1280	1020	0	3690	1950	1650	0	1840	0	0	6943
4	400	1711	1040	1280	1020	0	3690	1950	1650	0	1840	0	0	6849
5	400	2000	1040	1280	1020	0	3690	1950	1650	0	1840	0	0	6560
6	400	2300	1040	1280	1020	0	3690	1950	2518	0	1840	0	0	5392
7	400	2300	1520	1280	1400	0	3690	1950	3510	0	1840	0	0	3540
8	660	2300	1520	2033	1900	0	3690	1950	3510	0	1840	0	0	2027
9	660	2300	1520	2560	1900	0	3690	2317	3510	0	1840	0	0	1133
10	660	2300	1520	2560	1900	0	3690	2317	3510	0	1840	0	0	1133
11	660	2300	1520	2560	1900	0	3690	2058	3510	0	1840	0	0	1392
12	660	2300	1520	2560	1900	0	3690	2119	3510	0	1840	0	0	1331
13	660	2300	1520	2549	1900	0	3690	1950	3510	0	1840	0	0	1511
14	660	2300	1520	2306	1900	0	3690	1950	3510	0	1840	0	0	1755
15	660	2300	1520	2183	1900	0	3690	1950	3510	0	1840	0	0	1877
16	660	2300	1520	2377	1900	0	3690	1950	3510	0	1840	0	0	1683
17	660	2300	1520	2195	1900	0	3690	1950	3510	0	1840	0	0	1865
18	660	2300	1520	1737	1900	0	3690	1950	3510	0	1840	0	0	2323
19	660	2300	1520	1825	1900	0	3690	1950	3510	0	1840	0	0	2235
20	660	2300	1520	2660	1900	0	3650	2061	3510	0	1840	0	0	1389
21	660	2300	1520	2560	1900	0	3690	2237	3510	0	1840	0	0	1213
22	660	2300	1520	1777	1900	0	3690	1950	3510	0	1840	0	0	2283
23	400	2300	1520	1280	1328	0	3690	1950	3510	0	1840	0	0	3612
24	400	2300	1040	1280	1020	0	3690	1950	3303	0	1840	0	0	4607
Unit supply	13500	52656	33120	45861	38369	0	88560	48269	73741	0	44160	0	0	438175
Marginal cost	99242	266496	221926	340425	281824	0	232899	405278	491581	0	618240	0	0	2957910
Scenario 2	0.244201													
t	*x(1,t,2)*	*x(2,t,2)*	*x(3,t,2)*	*x(4,t,2)*	*x(5,t,2)*	*x(6,t,2)*	*x(7,t,2)*	*x(8,t,2)*	*x(9,t,2)*	*x(10,t,2)*	*x(11,t,2)*	*x(12,t,2)*	*x(13,t,2)*	*y(t,2)*

1	400	2269	1040	1280	1020	0	3690	1950	1650	0	1840	0	0	6291
2	400	2131	1040	1280	1020	0	3690	1950	1650	0	1840	0	0	6429
3	400	1996	1040	1280	1020	0	3690	1950	1650	0	1840	0	0	6561
4	400	2094	1040	1280	1020	0	3690	1950	1650	0	1840	0	0	6466
5	400	2300	1040	1280	1020	0	3690	1950	1742	0	1840	0	0	6168
6	400	2300	1040	1280	1020	0	3690	1950	2940	0	1840	0	0	4970
7	400	2300	1520	1280	1871	0	3690	1950	3510	0	1840	0	0	3069
8	660	2300	1520	2543	1900	0	3690	1950	3510	0	1840	0	0	1517
9	660	2300	1520	2560	1900	0	3690	2851	3510	0	1840	0	0	599
10	660	2300	1520	2560	1900	0	3690	2851	3510	0	1840	0	0	599
11	660	2300	1520	2560	1900	0	3690	2586	3510	0	1840	0	0	864
12	660	2300	1520	2560	1900	0	3690	2648	3510	0	1840	0	0	802
13	660	2300	1520	2560	1900	0	3690	2463	3510	0	1840	0	0	987
14	660	2300	1520	2560	1900	0	3690	2213	3510	0	1840	0	0	1237
15	660	2300	1520	2560	1900	0	3690	2088	3510	0	1840	0	0	1363
16	660	2300	1520	2560	1900	0	3690	2286	3510	0	1840	0	[illegible]	1164
17	660	2300	1520	2560	1900	0	3690	2100	3510	0	1840	0	0	1360
18	660	2300	1520	2240	1900	0	3690	1950	3510	0	1840	0	0	1820
19	660	2300	1520	2330	1900	0	3690	1950	3510	0	1840	0	0	1730
20	660	2300	1520	2560	1900	0	3690	2589	3510	0	1840	0	0	861
21	660	2300	1520	2560	1900	0	3690	2769	3510	0	1840	0	0	681
22	660	2300	1520	2281	1900	0	3690	1950	3510	0	1840	0	0	1779
23	400	2300	1520	1280	1797	0	3690	1950	3510	0	1840	0	0	3143
24	400	2300	1275	1280	1020	0	3690	1950	3510	0	1840	0	0	4165
Unit supply	13500	54493	33355	49075	39308	0	[illegible]	52793	74462	0	44160	0	0	449706
Marginal cost	99242	275796	223502	364279	286726	0	232899	443813	496388	0	618240	0	0	3042884

Scenario 3	0,40262													
t	*x(1,t,3)*	*x(2,t,3)*	*x(3,t,3)*	*x(4,t,3)*	*x(5,t,3)*	*x(6,t,3)*	*x(7,t,3)*	*x(8,t,3)*	*x(9,t,3)*	*x(10,t,3)*	*x(11,t,3)*	*x(12,t,3)*	*x(13,t,3)*	*y(t,3)*
1	400	2300	1040	1280	1020	0	3690	1950	2007	0	1840	0	0	5903
2	400	2300	1040	1280	1020	0	3690	1950	1866	0	1840	0	0	6044
3	400	2300	1040	1280	1020	0	3690	1950	1730	0	1840	0	0	6180
4	400	2300	1040	1280	1020	0	3690	1950	1828	0	1840	0	0	6082
5	400	2300	1040	1280	1020	0	3690	1950	2133	0	1840	0	0	5777
6	400	2300	1040	1280	1020	0	3690	1950	3362	0	1840	0	0	4548
7	660	2300	1520	1462	1900	0	3690	1950	3510	0	1840	0	0	2598
8	660	2300	1520	2560	1900	0	3690	2444	3510	0	1840	0	0	1006
9	660	2300	1520	2560	1900	0	3690	3385	3510	0	1840	0	0	65
10	660	2300	1520	2560	1900	0	3690	3385	3510	0	1840	0	0	65
11	660	2300	1520	2560	1900	0	3690	[illegible]	3510	0	1840	0	0	337
12	660	2300	1520	2560	1900	0	3690	3177	3510	0	1840	0	0	273

t	x(1,t,s)	x(2,t,s)	x(3,t,s)	x(4,t,s)	x(5,t,s)	x(6,t,s)	x(7,t,s)	x(8,t,s)	x(9,t,s)	x(10,t,s)	x(11,t,s)	x(12,t,s)	x(13,t,s)	y(t,s)
13	660	2300	1520	2560	1900	0	3690	2987	3510	0	1840	0	0	463
14	660	2300	1520	2560	1900	0	3690	2731	3510	0	1840	0	0	719
15	660	2300	1520	2560	1900	0	3690	2602	3510	0	1840	0	0	848
16	660	2300	1520	2560	1900	0	3690	2806	3510	0	1840	0	0	644
17	660	2300	1520	2560	1900	0	3690	2615	3510	0	1840	0	0	835
18	660	2300	1520	2560	1900	0	3690	2133	3510	0	1840	0	0	1317
19	660	2300	1520	2560	1900	0	3690	2225	3510	0	1840	0	0	1225
20	660	2300	1520	2560	1900	0	3690	3116	3510	0	1840	0	0	334
21	660	2300	1520	2560	1900	0	3690	3301	3510	0	1840	0	0	149
22	660	2300	1520	2560	1900	0	3690	2175	3510	0	1840	0	0	1275
23	660	2300	1520	1386	1900	0	3690	1950	3510	0	1840	0	0	2674
24	400	2300	1520	1280	1218	0	3690	1950	3510	0	1840	0	0	3722
Unit supply	14020	55200	33600	50208	39638	0	88560	59745	76106	0	44160	0	0	461237
Marginal cost	103064	279373	225142	372692	291148	0	232899	502256	607350	0	618240	0	0	3132162
Scenario 4	0.244201													
t	x(1,t,4)	x(2,t,4)	x(3,t,4)	x(4,t,4)	x(5,t,4)	x(6,t,4)	x(7,t,4)	x(8,t,4)	x(9,t,4)	x(10,t,4)	x(11,t,4)	x(12,t,4)	x(13,t,4)	y(t,4)
1	400	2300	1040	1280	1020	0	3690	1950	2395	0	1840	0	0	5515
2	400	2300	1040	1280	1020	0	3690	1950	2251	0	1840	0	0	5659
3	400	2300	1040	1280	1020	0	3690	1950	2111	0	1840	0	0	5799
4	400	2300	1040	1280	1020	0	3690	1950	2212	0	1840	0	0	5698
5	400	2300	1040	1280	1020	0	3690	1950	2524	0	1840	0	0	5386
6	400	2300	1314	1280	1020	0	3690	1950	3510	0	1840	0	0	4126
7	660	2300	1520	1933	1900	0	3690	1950	3510	0	1840	0	0	2127
8	660	2300	1520	2560	1900	0	3690	2955	3510	0	1840	0	0	495
9	660	2300	1520	2560	1900	0	3690	3450	3510	0	1840	342	127	-469
10	660	2300	1520	2560	1900	0	3690	3450	3510	0	1840	342	127	-469
11	660	2300	1520	2560	1900	0	3690	3450	3510	0	1840	190	0	-190
12	660	2300	1520	2560	1900	0	3690	3450	3510	0	1840	256	0	-256
13	660	2300	1520	2560	1900	0	3690	3450	3510	0	1840	61	0	-61
14	660	2300	1520	2560	1900	0	3690	3249	3510	0	1840	0	0	201
15	660	2300	1520	2560	1900	0	3690	3117	3510	0	1840	0	0	334
16	660	2300	1520	2560	1900	0	3690	3326	3510	0	1840	0	0	124
17	660	2300	1520	2560	1900	0	3690	3130	3510	0	1840	0	0	320
18	660	2300	1520	2560	1900	0	3690	2636	3510	0	1840	0	0	814
19	660	2300	1520	2560	1900	0	3690	2730	3510	0	1840	0	0	720
20	660	2300	1520	2560	1900	0	3690	3450	3510	0	1840	193	0	-193
21	660	2300	1520	2560	1900	0	3690	3450	3510	0	1840	342	41	-383
22	660	2300	1520	2560	1900	0	3690	2675	3510	0	1840	0	0	771
23	660	2300	1520	1655	1900	0	3690	1950	3510	0	1840	0	0	2205
24	400	2300	1520	1280	1661	0	3690	1950	3510	0	1840	0	0	3279

Unit supply	14020	55200	33874	51148	40081	0	88560	65520	[illegible]3	0	44160	1727	295	472768
Marginal cost	103064	279373	226978	379667	294400	0	232899	550806	[illegible]196	0	618240	850454	2952900	7009976

Scenario 5	0.054489													
t	*x(1,t,5)*	*x(2,t,5)*	*x(3,t,5)*	*x(4,t,5)*	*x(5,t,5)*	*x(6,t,5)*	*x(7,t,5)*	*x(8,t,5)*	*x(9,t,5)*	*x(10,t,5)*	*x(11,t,5)*	*x(12,t,5)*	*x(13,t,5)*	*y(t,5)*
1	400	2300	1040	1280	1020	0	3690	1950	2763	0	1840	0	0	5127
2	400	2300	1040	1280	1020	0	3690	1950	2635	0	1840	0	0	5275
3	400	2300	1040	1280	1020	0	3690	1950	2493	0	1840	0	0	5418
4	400	2300	1040	1280	1020	0	3690	1950	2595	0	1840	0	0	5315
5	400	2300	1040	1280	1020	0	3690	1950	2916	0	1840	0	0	4894
6	400	2300	1520	1280	1236	0	3690	1950	3510	0	1840	0	0	3704
7	660	2300	1520	2404	1900	0	3690	1950	3510	0	1840	0	0	1656
8	660	2300	1520	2560	1900	0	3690	3450	3510	0	1840	15	0	-15
9	660	2300	1520	2560	1900	0	3690	3450	3510	0	1840	342	661	-1003
10	660	2300	1520	2560	1900	0	3690	3450	3510	0	1840	342	661	-1003
11	660	2300	1520	2560	1900	0	3690	3450	3510	0	1840	342	376	-718
12	860	2300	1520	2560	1900	0	3690	3450	3510	0	1840	342	443	-785
13	660	2300	1520	2560	1900	0	3590	3450	3510	0	1840	342	243	-585
14	660	2300	1520	2560	1900	0	3690	3450	3510	0	1840	317	0	-317
15	660	2300	1520	2560	1900	0	3690	3450	3510	0	1840	181	0	-181
16	660	2300	1520	2560	1900	0	3690	3450	3510	0	1840	342	53	-395
17	660	2300	1520	2560	1900	0	3690	3450	3510	0	1840	195	0	-195
18	660	2300	1520	2560	1900	0	3690	3139	3510	0	1840	0	0	311
19	660	2300	1520	2560	1900	0	3690	3235	3510	0	1840	0	0	215
20	660	2300	1520	2560	1900	0	3690	3450	3510	0	1840	342	379	-721
21	660	2300	1520	2560	1900	0	3690	3450	3510	0	1840	342	573	-915
22	660	2300	1520	2560	1900	0	3690	3183	3510	0	1840	0	0	267
23	660	2300	1520	2324	1900	0	3690	1950	3510	0	1840	0	0	1736
24	603	2300	1520	1280	1900	0	3690	1950	3510	0	1840	0	0	2837
Unit supply	14223	55200	34080	52087	40536	0	88560	68507	80112	0	44160	3444	3390	484299
Marginal cost	104559	279373	228358	386643	297745	0	232899	575911	534057	0	618240	1695973	33895000	36848756

Expected marginal	102004	277798	225190	371343	291202	0	232899	498569	508650	0	618240	300093	2567996	5993983
Total unit cost	114983	303491	264434	416853	329554	0	258804	559811	574680	0	618240	300093	2567996	6308938

i	1	2	3	4	5	6	7	8	9	10	11	12	13	
Unit	Arnot	Duvha	Hendrina	Kendal	Kriel	Lethabo	Matimba	Matla	Tutuka	Majuba	Koeberg	Gas	Unmet	Sum
u'(i)	2	4	7	4	4	4	6	6	6	0	2	0	0	
u(i)	2	4	7	4	4	3	6	6	6	0	2	0	0	
v(i)	0	0	0	0	0	0	0	0	0	0	0	0	0	
g(i)u(i)	400	1400	910	1280	1020	1125	2160	1950	1650	0	1840	0	0	Capacity
G(i)u(i)	660	2300	1330	2560	1900	1779	3690	3450	3510	0	1840	0	∞	23079
Switch-on cost	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Fixed cost	12979	25693	19026	45510	38352	20942	25906	61242	66030	0	0	0	0	315679
Scenario 1	0.054489													
t	*x(1,t,1)*	*x(2,t,1)*	*x(3,t,1)*	*x(4,t,1)*	*x(5,t,1)*	*x(6,t,1)*	*x(7,t,1)*	*x(8,t,1)*	*x(9,t,1)*	*x(10,t,1)*	*x(11,t,1)*	*x(12,t,1)*	*x(13,t,1)*	*y(t,1)*
1	400	1400	910	1280	1020	1125	3176	1950	1650	0	1840	0	0	8268
2	400	1400	910	1280	1020	1125	3042	1950	1650	0	1840	0	0	8402
3	400	1400	910	1280	1020	1125	2913	1950	1650	0	1840	0	0	8532
4	400	1400	910	1280	1020	1125	3006	1950	1650	0	1840	0	0	8438
5	400	1400	910	1280	1020	1125	3295	1950	1650	0	1840	0	0	8149
6	400	2173	910	1280	1020	1125	3690	1950	1650	0	1840	0	0	6981
7	400	2300	910	1280	1020	1125	3690	1950	3375	0	1840	0	0	5129
8	478	2300	1330	1280	1900	1125	3690	1950	3510	0	1840	0	0	3616
9	660	2300	1330	1992	1900	1125	3690	1950	3510	0	1840	0	0	2722
10	660	2300	1330	1992	1900	1125	3690	1950	3510	0	1840	0	0	2722
11	660	2300	1330	1733	1900	1125	3690	1950	3510	0	1840	0	0	2981
12	660	2300	1330	1794	1900	1125	3690	1950	3510	0	1840	0	0	2920
13	660	2300	1330	1614	1900	1125	3690	1950	3510	0	1840	0	0	3100
14	660	2300	1330	1371	1900	1125	3690	1950	3510	0	1840	0	0	3344
15	628	2300	1330	1280	1900	1125	3690	1950	3510	0	1840	0	0	3466
16	660	2300	1330	1442	1900	1125	3690	1950	3510	0	1840	0	0	3272
17	640	2300	1330	1280	1900	1125	3690	1950	3510	0	1840	0	0	3454
18	400	2300	1330	1280	1682	1125	3690	1950	3510	0	1840	0	0	3912
19	400	2300	1330	1280	1770	1125	3690	1950	3510	0	1840	0	0	3824
20	660	2300	1330	1736	1900	1125	3690	1950	3510	0	1840	0	0	2978
21	660	2300	1330	1912	1900	1125	3690	1950	3510	0	1840	0	0	2802
22	400	2300	1330	1280	1722	1125	3690	1950	3510	0	1840	0	0	3872
23	400	2300	910	1280	1020	1125	3690	1950	3303	0	1840	0	0	5201
24	400	2300	910	1280	1020	1125	3690	1950	2308	0	1840	0	0	6196
Unit supply	12486	50573	28140	34785	37154	27000	85641	46800	71536	0	44160	0	0	438175
Marginal cost	91787	255955	188566	258209	272905	253216	224959	393431	476886	0	618240	0	0	3034143
Scenario 2	0.244201													
t	*x(1,t,2)*	*x(2,t,2)*	*x(3,t,2)*	*x(4,t,2)*	*x(5,t,2)*	*x(6,t,2)*	*x(7,t,2)*	*x(8,t,2)*	*x(9,t,2)*	*x(10,t,2)*	*x(11,t,2)*	*x(12,t,2)*	*x(13,t,2)*	*y(t,2)*

1	400	1400	910	1280	1020	1125	3564	1950	1650	0	1840	0	0	7880
2	400	1400	910	1280	1020	1125	3426	1950	1650	0	1840	0	0	8018
3	400	1400	910	1280	1020	1125	3294	1950	1650	0	1840	0	0	8150
4	400	1400	910	1280	1020	1125	3389	1950	1650	0	[illegible]	0	0	8055
5	400	1400	910	1280	1020	1125	3687	1950	1650	0	1840	0	0	7757
6	400	2300	910	1280		1125	3690	1950	1945	0	1840	0	0	6559
7	400	2300	1246	1280		1125	3690	1950	3510	0	1840	0	0	4658
8	660	2300	1330	1608	1900	1125	3690	1950	3510	0	1840	0	0	3106
9	660	2300	1330	2526	1900	1125	3690	1950	3510	0	1840	0	0	2188
10	660	2300	1330	2526	1900	1125	3690	1950	3510	0	1840	0	0	2188
11	660	2300	1330	2261	1800	1125	3690	1950	3510	0	1840	0	0	2453
12	660	2300	1330	2323	1900	1125	3690	1950	3510	0	1840	0	0	2391
13	660	2300	1330	2138	1900	1125	3690	1950	3510	0	1840	0	0	2576
14	660	2300	1330	1888	1900	1125	3690	1950	3510	0	1840	0	0	2826
15	660	2300	1330	1763	1900	1125	3690	1950	3510	0	1840	0	0	2952
16	660	2300	1330	1961	1900	1125	3690	1950	3510	0	1840	0	0	2753
17	660	2300	1330	1775	1900	1125	3690	1950	3510	0	1840	0	0	2939
18	660	2300	1330	1305	1900	1125	3690	1950	3510	0	1840	0	0	3409
19	660	2300	1330	1395	1900	1125	3690	1950	3510	0	1840	0	0	3319
20	660	2300	1330	2264	1900	1125	3690	1950	3510	0	1840	0	0	2450
21	660	2300	1330	2444	1900	1125	3690	1950	3510	0	1840	0	0	2270
22	660	2300	1330	1346	1900	1125	3690	1950	3510	0	1840	0	0	3368
23	400	2300	1172	1280	1020	1125	3690	1950	3510	0	18[illegible]0	0	0	4732
24	400	2300	910	1280	1020	1125	3690	1950	2750	0	1840	0	0	5754
Unit supply	13500	50700	28738	41043	37680	27000	87470	46800	72615	0	44160	0	0	449706
Marginal cost	99242	256598	192565	304658	276766	253216	230032	393431	484080	0	618240	0	0	3108827

Scenario 3	0.40262													
t	*x(1,t,3)*	*x(2,t,3)*	*x(3,t,3)*	*x(4,t,3)*	*x(5,t,3)*	*x(6,t,3)*	*x(7,t,3)*	*x(8,t,3)*	*x(9,t,3)*	*x(10,t,3)*	*x(11,t,3)*	*x(12,t,3)*	*x(13,t,3)*	*y(t,3)*
1	400	1662	910	1280	1020	1125	3690	1950	1650	0	1840	0	0	7492
2	400	1521	910	1280	1020	1125	3690	1950	1650	0	1840	0	0	7633
3	400	1400	910	1280	1020	1125	3675	1950	1650	0	1840	0	0	7769
4	400	1483	910	1280	1020	1125	3690	1950	1650	0	1840	0	0	7671
5	400	1788	910	1280	1020	1125	3690	1950	1650	0	1840	0	0	7366
6	400	2300	910	1280	1020	1125	3690	1950	2367	0	1840	0	0	6137
7	400	2300	1330	1280	1407	1125	3690	1950	3510	0	1840	0	0	4187
8	660	2300	1330	2119	1900	1125	3690	1950	3510	0	1840	0	0	2595
9	660	2300	1330	2560	1900	1125	3690	2450	3510	0	1840	0	0	1654
10	660	2300	1330	2560	1900	1125	3690	2450	3510	0	1840	0	0	1654
11	660	2300	1330	2560	1900	1125	3690	2178	3510	0	1840	0	0	1926
12	660	2300	1330	2560	1900	1125	3690	2242	3510	0	1840	0	0	1862

13	660	2300	1330	2560	1900	1125	3690	2052	3510	0	1840	0	0	2052
14	660	2300	1330	2406	1900	1125	3690	1950	3510	0	1840	0	0	2308
15	660	2300	1330	2277	1900	1125	3690	1950	3510	0	1840	0	0	2437
16	660	2300	1330	2481	1900	1125	3690	1950	3510	0	1840	0	0	2233
17	660	2300	1330	2290	1900	1125	3690	1950	3510	0	1840	0	0	2424
18	660	2300	1330	1808	1900	1125	3690	1950	3510	0	1840	0	0	2906
19	660	2300	1330	1900	1900	1125	3690	1950	3510	0	1840	0	0	2814
20	660	2300	1330	2560	1900	1125	3690	2181	3510	0	1840	0	0	1923
21	660	2300	1330	2560	1900	1125	3690	2366	3510	0	1840	0	0	1738
22	660	2300	1330	1850	1900	1125	3690	1950	3510	0	1840	0	0	2864
23	400	2300	1330	1280	1331	1125	3690	1950	3510	0	1840	0	0	4263
24	400	2300	910	1280	1020	1125	3690	1950	3193	0	1840	0	0	5311
Unit supply	13500	51554	28980	46571	38378	27000	88545	49069	73480	0	44160	0	0	461237
Marginal cost	99242	260920	194185	345695	281893	253216	232859	412505	489844	0	618240	0	0	3188598

Scenario 4	0.244201													
t	*x(1,t,4)*	*x(2,t,4)*	*x(3,t,4)*	*x(4,t,4)*	*x(5,t,4)*	*x(6,t,4)*	*x(7,t,4)*	*x(8,t,4)*	*x(9,t,4)*	*x(10,t,4)*	*x(11,t,4)*	*x(12,t,4)*	*x(13,t,4)*	*y(t,4)*
1	400	2050	910	1280	1020	1125	3690	1950	1650	0	1840	0	0	7104
2	400	1906	910	1280	1020	1125	3690	1950	1650	0	1840	0	0	7248
3	400	1766	910	1280	1020	1125	3690	1950	1650	0	1840	0	0	7388
4	400	1867	910	1280	1020	1125	3690	1950	1650	0	1840	0	0	7287
5	400	2179	910	1280	1020	1125	3690	1950	1650	0	1840	0	0	6975
6	400	2300	910	1280	1020	1125	2690	1950	2789	0	1840	0	0	5715
7	400	2300	1330	1280	1878	1125	3690	1950	3510	0	1840	0	0	3716
8	660	2300	1330	2560	1900	1125	3690	2020	3510	0	1840	0	0	2084
9	660	2300	1330	2560	1900	1125	3690	2984	3510	0	1840	0	0	1120
10	660	2300	1330	2560	1900	1125	3690	2984	3510	0	1840	0	0	1120
11	660	2300	1330	2560	1900	1125	3690	2705	3510	0	1840	0	0	1399
12	660	2300	1330	2560	1900	1125	3690	2771	3510	0	1840	0	0	1333
13	660	2300	1330	2560	1900	1125	3690	2576	3510	0	1840	0	0	1528
14	660	2300	1330	2560	1900	1125	3690	2314	3510	0	1840	0	0	1790
15	660	2300	1330	2560	1900	1125	3690	2182	3510	0	1840	0	0	1923
16	660	2300	1330	2560	1900	1125	3690	2391	3510	0	1840	0	0	1713
17	660	2300	1330	2560	1900	1125	3690	2195	3510	0	1840	0	0	1909
18	660	2300	1330	2311	1900	1125	3690	1950	3510	0	1840	0	0	2403
19	660	2300	1330	2405	1900	1125	3690	1950	3510	0	1840	0	0	2309
20	660	2300	1330	2560	1900	1125	3690	2708	3510	0	1840	0	0	1396
21	660	2300	1330	2560	1900	1125	3690	2898	3510	0	1840	0	0	1206
22	660	2300	1330	2354	1800	1125	3690	1950	3510	0	1840	0	0	2360
23	400	2300	1330	1280	1800	1125	3690	1950	3510	0	1840	0	0	3794
24	400	2300	1035	1280	1020	1125	3690	1950	3510	0	1840	0	0	4868

Unit supply	13500	53468	29106	49310	39318	27000	88560	54128	74219	0	44160	0	0	472768
Marginal cost	99242	270607	195027	366025	288795	253216	232899	455030	494771	0	618240	0	0	3273851
Scenario 5	0.054489													
t	*x(1,t,5)*	*x(2,t,5)*	*x(3,t,5)*	*x(4,t,5)*	*x(5,t,5)*	*x(6,t,5)*	*x(7,t,5)*	*x(8,t,5)*	*x(9,t,5)*	*x(10,t,5)*	*x(11,t,5)*	*x(12,t,5)*	*x(13,t,5)*	*y(t,5)*
1	400	2300	910	1280	1020	1125	3690	1950	1788	0	1840	0	0	6716
2	400	2290	910	1280	1020	1125	3690	1950	1650	0	1840	0	0	6864
3	400	2148	910	1280	1020	1125	3690	1950	1650	0	1840	0	0	7007
4	400	2250	910	1280	1020	1125	3690	1950	1650	0	1840	0	0	6904
5	400	2300	910	1280	1020	1125	3690	1950	1927	0	1840	0	0	6583
6	400	2300	910	1280	1020	1125	3690	1950	3211	0	1840	0	0	5293
7	660	2300	1330	1469	1900	1125	3690	1950	3510	0	1840	0	0	3245
8	660	2300	1330	2560	1900	1125	3690	2530	3510	0	1840	0	0	1574
9	660	2300	1330	2560	1900	1193	3690	3450	3510	0	1840	0	0	586
10	660	2300	1330	2560	1900	1191	3690	3450	3510	0	1840	0	0	586
11	660	2300	1330	2560	1900	1125	3690	3233	3510	0	1840	0	0	871
12	660	2300	1330	2560	1900	1125	3690	3300	3510	0	1840	0	0	804
13	660	2300	1330	2560	1900	1125	3690	3100	3510	0	1840	0	0	1004
14	660	2300	1330	2560	1900	1125	3690	2832	3510	0	1840	0	0	1273
15	660	2300	1330	2560	1900	1125	3690	2696	3510	0	1840	0	0	1408
16	660	2300	1330	2560	1900	1125	3690	2910	3510	0	1840	0	0	1194
17	660	2300	1330	2560	1900	1125	3690	2710	3510	0	1840	0	0	1394
18	660	2300	1330	2560	1900	1125	3690	2204	3510	0	1840	0	0	1900
19	660	2300	1330	2560	1900	1125	3690	2300	3610	0	1840	0	0	1804
20	660	2300	1330	2560	1900	1125	3690	3236	3510	0	1840	0	0	868
21	660	2300	1330	2560	1900	1125	3690	3430	3510	0	1840	0	0	674
22	660	2300	1330	2560	1900	1125	3690	2248	3510	0	1840	0	0	1856
23	660	2300	1330	1389	1900	1125	3690	1950	3510	0	1840	0	0	3225
24	400	2300	1330	1280	1168	1125	3690	1950	3510	0	1840	0	0	4426
Unit supply	14020	54988	29400	50217	39588	27136	88560	61178	75050	0	44160	0	0	484299
Marginal cost	103054	278301	196999	372762	290784	254495	232899	514303	500312	0	618240	0	0	3362158
Expected marginal	99044	262907	193842	337346	282321	253285	231750	422739	489504	0	618240	0	0	3190978
Total unit cost	112023	288599	212868	382856	320673	274228	257656	483962	555534	0	618240	0	0	3506687

Bibliography

[1] ABB Systems Control (1996), *Couger User's Manual, Version 6.37*. ABB Power T&D Company Inc., Santa Clara, California.

[2] N.H. Agnew, R.A. Agnew, J. Rasmussen & K.R. Smith (1969), "An application of chance-constrained programming to portfolio selection in a casualty insurance firm." *Management Science* 15B, 512-520.

[3] H. Anton (1991), *Elementary Linear Algebra* (6/e). Wiley, New York.

[4] M. Avriel & A. Williams (1970), "The value of information and stochastic programming." *Operations Research* 18, 947-954.

[5] J.F. Benders (1962), "Partitioning procedures for solving mixed-variable programming problems." *Numerische Mathematik* 4, 238-252.

[6] J.R. Birge (1982), "The value of the stochastic solution in stochastic linear programs with fixed recourse." *Mathematical Programming* 24, 314-325.

[7] J.R. Birge (1995), "Models and model value in stochastic programming." *Annals of Operations Research* 59, 1-18.

[8] J.R. Birge & C.H. Rosa (1995), "Modelling investment uncertainty in the costs of global CO_2 emission policy." *European Journal of Operational Research* 83, 466-488.

[9] J.R. Birge & F.V. Louveaux (1988), "A multicut algorithm for two-stage stochastic linear programs." *European Journal of Operational Research* 34 384-392.

[10] J.R. Birge & F.V. Louveaux (1997), *Introduction to Stochastic Programming.* Springer-Verlag, Berlin.

[11] C. Borell (1975), "Convex set functions in d-spaces." *Periodica Mathematica Hungarica* 6, 111-136.

[12] CIGRE Task Force 38-04-01 (1997), "Unit commitment modelling and optimization." Draft for publication in ELECTRA by Patrick Sandrin, April 1997.

[13] G. Consigli & M.A.H. Dempster (1996), "Dynamic stochastic programming for asset-liability management." To appear in *Annals of Operations Research*, Proceedings of APMOD95 Conference, Brunel University, UK.

[14] G.B. Dantzig (1963), *Linear Programming and Extensions.* Princeton University Press, Princeton, New Jersey.

[15] G.B. Dantzig & P. Wolfe (1960), "The decomposition principle for linear programs." *Operations Research* 8, 101-111.

[16] A. de G. Brandão & W. Zucchini (1990), *A Stochastic Daily Climate Model for South African Conditions.* Water Research Commission Report No. 200/1/90, ISBN 0947447792.

[17] J. Dean, Eskom (National Control - System Operations), Personal Communication.

[18] M.A.H. Dempster (ed) (1980) *Stochastic Programming.* Academic Press, London.

[19] J. Dupačová, A. Gaivoronski, Z. Kos & T. Szántai (1991), "Stochastic programming in water management: a case study and a comparison of solution techniques." *European Journal of Operational Research* 52, 28-44.

[20] L.F. Escudero, P.V. Kamesam, A.J. King & R.J-B. Wets (1993), "Production planning via scenario modelling." *Annals of Operations Research* 43, 311-335.

[21] A. Ferguson & G.B. Dantzig (1956), "The allocation of aircraft to routes: an example of linear programming under uncertain demands." *Management Science* 3, 45-73.

[22] H.I. Gassmann (1989), "Optimal harvest of a forest in the presence of uncertainty." *Canadian Journal of Forest Research* 19, 1267-1274.

[23] H.I. Gassmann (1990), "MSLiP: a computer code for the multistage stochastic linear programming problem." *Mathematical Programming* 47, 407-423.

[24] P.E. Gill, W. Murray, M.A. Saunders & M.H. Wright (1989), "Constrained Nonlinear Programming." In G.L. Nemhauser *et al.* (eds), *Handbooks in OR & MS, Volume 1: Optimization*, 171-210, Elsevier, Amsterdam.

[25] J. Higle & S. Sen (1991), "Stochastic decomposition: an algorithm for two stage linear programs with recourse." *Mathematics of Operations Research* 16, 650-669.

[26] D. Holmes (1995), "What is stochastic programming?" Internet address at http://www-personal.umich.edu/~jrbirge/dholmes/StoProIntro.html.

[27] P. Kall (1976), *Stochastic Linear Programming*. Springer-Verlag, Berlin.

[28] P. Kall & S.W. Wallace (1994), *Stochastic Programming*. Wiley, Chichester.

[29] E. Kao & M. Queyranne (1985), "Budgeting costs of nursing in a hospital." *Management Science* 31, 608-621.

[30] M.I. Kusy & W.T. Ziemba (1986), "A bank asset and liability management model." *Operations Research* 34, 356-376.

[31] S.R. Lay (1982), *Convex Sets and Their Applications*. Wiley, New York.

[32] C. Lemaréchal (1989), "Nondifferentiable Optimization." In G.L. Nemhauser *et al.* (eds), *Handbooks in OR & MS, Volume 1: Optimization*, 529-572, Elsevier, Amsterdam.

[33] F.V. Louveaux & Y. Smeers (1997), "Stochastic optimization for the introduction of a new energy technology." *Stochastics* (to appear).

[34] D.G. Luenberger (1973), *Introduction to Linear and Nonlinear Programming*. Addison-Wesley, Reading, Massachusetts.

[35] W.H. Marlow (1978), *Mathematics for Operations Research*. Wiley, New York.

[36] The MathWorks Inc. (1997), *Optimization Toolbox User's Guide for Matlab 5*. The MathWorks Inc., Natick, Massachusetts.

[37] Optimization Technology Centre (1997), "Natural gas planning: an application of stochastic linear programming." Internet address http://www.mcs.anl.gov/home/otc/Guide/CaseStudies/slp.

[38] A. Prékopa (1971), "Logarithmic concave measures with application to stochastic programming." *Acta. Sci. Math. (Szeged)* 32, 301-316.

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

[40] K. Schittowski (1985), "NLPQPL: A Fortran subroutine solving constrained nonlinear programming problems." *Annals of Operations Research*, 5, 485-500.

[41] J.K. Sengupta (1972), *Stochastic Programming: Methods and Applications.* North-Holland, Amsterdam.

[42] L. Stougie (1987), *Design and Analysis of Algorithms for Stochastic Integer Programmming.* Centrum voor Wiskunde en Informatica, Amsterdam.

[43] H.A. Taha (1975), *Integer Programming: Theory, Applications and Computations.* Academic Press, New York.

[44] H.A. Taha (1992), *Operations Research: an Introduction* (5/e). Macmillan, New York.

[45] S. Takriti, J.R. Birge & E. Long (1996), "A stochastic model of the unit commitment problem." *IEEE Transactions on Power Systems* 11, 3, 1497-1508.

[46] S. Vajda (1972), *Probabilistic Programming.* Academic Press, New York.

[47] M.H. van der Vlerk (1997), "Stochastic programming bibliography." Internet address http://mally.eco.rug.nl/biblio/stoprog.html.

[48] R. van Slyke & R.J-B. Wets (1969), "L-shaped linear programs with application to optimal control and stochastic programming." *SIAM Journal on Applied Mathematics* 17, 638-663.

[49] S.A. Zenios (1992), *Financial Optimization.* Cambridge University Press, Cambridge.

Author: Corrigall, Stuart.
Name of thesis: Stochastic programs and their value over deterministic programs.

PUBLISHER:
University of the Witwatersrand, Johannesburg
©2015

LEGALNOTICES:

Copyright Notice: All materials on the University of the Witwatersrand, Johannesburg Library website are protected by South African copyright law and may not be distributed, transmitted, displayed or otherwise published in any format, without the prior written permission of the copyright owner.

Disclaimer and Terms of Use: Provided that you maintain all copyright and other notices contained therein, you may download material (one machine readable copy and one print copy per page)for your personal and/or educational non-commercial use only.

The University of the Witwatersrand, Johannesburg, is not responsible for any errors or omissions and excludes any and all liability for any errors in or omissions from the information on the Library website.