To the Graduate Council:

I am submitting herewith a dissertation written by Guey-Mei You entitled "New Solution Methodologies for Multi-period Stochastic Programming Decision Models." I have examined the final copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Management Science.

N.C.P. Edirisinghe, Major Professor

We have read this dissertation
and recommend its acceptance:

Accepted for the Council:

Associate Vice Chancellor
and Dean of The Graduate School

NEW SOLUTION METHODOLOGIES FOR MULTI-PERIOD STOCHASTIC PROGRAMMING DECISION MODELS

A Dissertation
Presented for the
Doctor of Philosophy
Degree
The University of Tennessee, Knoxville

Guey-Mei You

May 1999

To my mother and the memory of my father

Acknowledgments

I would like to thank my major professor, Chanaka Edirisinghe, for his guidance and continuous encouragement throughout this study. I would also like to thank my committee members, Ham Bozdogan, Charles Noon, M. Srinivasan, and Carl Wagner, who read over the manuscript and made suggestions. I am grateful for the financial support received from Management Science Program and Computing and Academic Services (CAS) during my doctoral study. I also wish to thank the Joint Institute for Computational Science (JICS) for providing a computing account which allow me to perform computations on the UTK SP2. Finally, I would like to give my thanks to my husband, Ching-Hsiang Wang, for his support, and to my sons, Ying-An and Jae-An, for their inspiration.

Abstract

Multistage stochastic programming is a tool for modeling dynamic sequential decision making problems which involve making a sequence of decisions under uncertainty of some future events. Such models have been successfully used for many applications including planning problems in finance, economics, production, engineering, and agriculture. We focus on models in which uncertainty is described by a set of discrete scenarios associated with known probabilities. The computational difficulty of such models is mainly due to the large number of future scenarios that need to be analyzed in the process of determining implementable decisions sequentially over time.

In Edirisinghe (1996) and Edirisinghe and You (1996), a scenario approximation method was proposed and its use in solving two-stage stochastic programs effectively was demonstrated. The two-stage scenario approximation method utilizes the second-order moment information and simplicial domains of the scenario space to determine representative scenario clusters. The resulting problems are much smaller in size and provide lower and upper approximations. By successively partitioning the scenario space, the approximations can be iteratively refined to a desired accuracy. We extend this method to solve multistage stochastic programs.

In the multistage case, we apply the scenario approximation procedure to the scenario space spanning the entire decision horizon. In this way, the number of approximating sce-

narios is small. However, the solutions converge to that for the two-stage equivalent of the multistage problem, which is the true multistage problem without nonanticipativity constraints. By restricting the technology matrices (or the tender information) to be non-stochastic in the multistage models, we derive appropriate nonanticipativity constraints from the relationships between the true multistage model and the approximating ones. Iterative approximation coupled with the derived nonanticipativity constraints provide a valid methodology for solving multistage stochastic programming model. Decomposition algorithms are designed to solve the problem more efficiently.

Contents

List of Figures

Chapter 1

Introduction

Mathematical programming provides an elegant approach for modeling complex decision problems arising in real-world applications. In mathematical programming, an optimization model is formulated based on the collected data pertaining to the underlying decision problem. Usually, it is assumed that the input data used in a mathematical optimization model is exactly known or accurately estimated. However, in a world punctuated with uncertainty, this need not be always the case and some degree of uncertainty about the input data exists. For example, demands for manufactured products, prices of stocks and interest rates in financial markets, and inflows into hydroelectric power plants are often random in nature. Decisions made without considering all (extreme) possibilities of uncertain events may lead to costly results in the future such as bankruptcies or floods.

Considering uncertainty of input data explicitly in mathematical programming models leads us to *stochastic programming models*. Stochastic programming originated in the early 1950's. The pioneering works include Dantzig [12], Beale [3], Tintner [69], and Charnes and Cooper [11]. Refer to Appendix A (Stochastic Programming's Orchard,1998) for the researchers in stochastic programming areas. In a stochastic programming model, random elements may be present either in the objective function, in the right-hand side, or in the

constraint matrix to reflect the inherent uncertainty in prices, demands, and other technology information. For instance, cash and portfolio management models involve stochasticity in the constraint matrix resulting from unknown future exchange rates and unknown rates of return. Problems in manufacturing, transportation, and power generation typically involve stochastic right_hand side elements that reflect uncertainty in supply and demand. The random variables representing uncertainty are usually assumed to have a known joint distribution which could be continuous or discrete.

Stochastic programming models can be classified into two types: *stochastic programs with recourse* and *stochastic programs with probabilistic (chance) constraints.* Both types of models are for 'here and now' situations in which the decisions are chosen prior to the knowledge of the random data. In chance-constrained programming, decisions are chosen so that constraints involving random input data are satisfied with a prescribed tolerance probability or reliability. See Edirisinghe et al [20] for an application of chance-constrained programming to water resource planning. For models and recent solution techniques of chance-constrained programming, we refer to Charnes and Cooper [11] and Prèkopa [62]. In the present work, our focus is on the former type of stochastic programming models, i.e., stochastic programming recourse models, which not only take into account *risk* in the problem formulation but also allow the possibility for *information gathering* during the decision process. For an overview of stochastic programming in general, refer to Dempster [15], Kall [45], Ermoliev and Wets [28], and Birge and Wets [9].

Stochastic programming with recourse is a tool for modeling multi-period decision problems with random data, such as production planning under uncertainty of future demands, and portfolio management under uncertainty of both future interest rates and funds avail-

able for investment. In addition to taking into account all the possible outcomes of uncertain events, another essential character of stochastic programming recourse models is that decision activities are divided into two or more stages. A first stage decision is the present decision to be determined with the expectation of all possible outcomes of future uncertain events but without the knowledge of the actual outcome of the uncertain events. The second (or higher) decision stage is a decision opportunity for making corrective decisions after the previous stage decision is made and after the outcome of the uncertain events occurring at that stage is known.

Consider the following capacity expansion problem in steel production. Due to increases in steel consumption, a steel company faces the need for building new capacity, if it wants to maintain its share of the market. The production capacity can be expanded either by building a plant today or by making only minor modifications in existing facilities. The facts are that the degree of increase of steel consumption and the the availability of new technology in the future are both uncertain. A decision to build a plant today with the current (existing) technology may mean in effect that for the next fifteen to twenty years the company cannot make a transition into the new technology except at prohibitive cost. On the other hand, by making the decision to postpone building, it may not be able to meet the demands if the increase in steel consumption is high and may foreclose other opportunities such as market position, perhaps irrevocably. In this example, it is thus important to consider all the possible future situations in the decision making. At the fist stage, decisions to be made are the steel production level to meet uncertain demand subject to the current capacity constraints and the need for building new capacity. After the first stage decisions have been made, information about steel consumption and about

new technology is gathered, and then a second stage recourse decision (either back-order or inventory) is made to correct any discrepancy which could be shortage or surplus of steel production due to the first stage decision. If, in the given time span of planning, another decision stage (or the third decision stage) is allowed, then the second stage decision is not only a recourse but also a risk-taking decision (adjusting production level and determining the need for new capacity) that takes into account possible increase of steel consumption and availability of new technology in the next stage. The same decision process (gathering information about uncertainties and making recourse decisions) is repeated successively for the subsequent decision stages. The objective in this example is then to make decisions which minimize the expected net cost or maximize the expected net profit. Based on the number of decision stages allowed in the planning time horizon, stochastic programming recourse models can be further classified into *two-stage* and *multi-stage* models.

Stochastic programming recourse models have been successfully applied in diverse areas of decision problems including production planning, energy planning, financial planning, engineering, and agriculture, as they capture the dynamic, sequential, and stochastic characters inherent in this class of decision problems. The Russell-Yasuda Kasai model, an asset-liability model for a Japanese insurance company, is an example of successful application of multi-stage stochastic programming model which aided to yield "extra income of 42 basis points or $ 79 million during the first two years of use", as reported in Cariño et al [10].

The major difficulty in incorporating stochastic programming models in a real-world decision process has been the computational complexity involved in solving such models. Complex stochastic feature in the objective function or constraint functions is one major

cause of computational difficulty encountered in solving stochastic models. For example, if the objective function is a function of random variables having continuous distributions, the evaluation of the expected value of the objective function involves multidimensional integration which requires computational efforts increasing rapidly with the dimension of random variables. More difficulties arise if the optimization technique requires the computation of gradients. The second major cause of solution difficulty is due to problem size. In the case that random variables are discrete (or discretized), the resultant stochastic programming recourse model turns out to be a large-scale mathematical programming problem. Solution methods using mathematical decomposition techniques ([70] [5] [37] [68]) seem promising but limited to problems of modest sizes.

Approximation techniques have been used to overcome the difficulties in solving stochastic programming recourse models involving either continuous or discrete distributions. There are two main approximation approaches: sampling-based approximation methods and bounds-based approximation methods. Sampling-based approximation methods ([26] [39] [14]) use sampling to obtain statistical estimates such as sub-gradients for approximating some functions. Such sampling-based methods are usually limited by a lack of computable bounds. Bounds-based approximation methods ([50]) replace the original problem with a simpler one by approximating the given distribution by discrete approximating distributions involving only a small number of points. The resultant approximating problems are much smaller in size and can be solved with relative ease. In addition, such approximations may be refined sequentially until the problem is solved to a prescribed accuracy.

In this research, the approach of bounds-based approximations is adopted to solve multi-period stochastic programming recourse models. The second-order simplicial approxima-

tions developed by Edirisinghe [19] serves as a basis for the proposed new methodologies. This research deals with the development of simplicial refinement schemes and new approximations for multi-stage models, the design of decomposition algorithms for efficient computations, and the implementation of the new solution methods for solving two-stage and multi-stage models.

The remainder of this dissertation is organized as follows. Preliminary definitions of some terms and concepts used throughout this thesis are provided in Appendix B. In Chapter 2, stochastic programming recourse model formulation for two-stage and multi-stage models are presented along with an overview of model properties. Chapter 3 provides a survey of existing solution approaches for solving stochastic programming recourse problems. Research objectives are then specified. To lay the groundwork for our methodologies, Chapter 4 provides a review of second-order simplicial approximation followed by a presentation of simplicial refinement schemes for solving two-stage stochastic linear programs. In Chapter 5, the proposed solution methodologies for solving multi-stage stochastic linear programs are presented. The implementation of the solution procedure and some computational results are also in Chapter 5. Chapter 5 concludes with a discussion on directions for future research.

Chapter 2

Overview of Stochastic Programming Recourse Models

Multi-period stochastic decision problems can be formulated as two-stage stochastic programming recourse models or multi-stage stochastic programming recourse models. The number of decision stages in a stochastic programming recourse model does not need to agree with the number of time periods considered in a decision problem. The former is defined by the grouping of decision activities (collecting information about uncertainty and making recourse decisions), while the latter is determined by the time horizon planned in the problem.

2.1 Two-stage models

In a two-stage stochastic decision problem, a first-stage decision x is made prior to any knowledge of the outcome of random events, then the outcome of random events is realized, and a second-stage recourse decision y is made.

2.1.1 Model formulations

A formulation of this decision problem is the following minimum cost two-stage stochastic linear program (SLP_2):

$$\begin{array}{lll} \min_x & \{\ c\,x & + E_\omega\,\phi(x,\omega)\ \} \\ \text{s.t.} & A\,x & = b \\ & x & \geq 0, \end{array} \tag{2.1}$$

where $c \in \Re^n$, $b \in \Re^m$, A is a fixed matrix of dimension $m \times n$, and the objective function comprises cx the cost incurred by the first stage decision and $E_\omega\phi(x,\omega)$ the expected value of the *recourse function* $\phi(x,\omega)$. The operator E_ω denotes mathematical expectation with respect to random vector ω. The recourse function $\phi(x,\omega)$ is a second stage cost function which depends on the first stage decision x and the realization of random vector ω and is determined by the second stage decision (recourse) problem:

$$\begin{array}{lllll} \phi(x,\omega) := & \min_y & q(\eta)\,y & & \\ & \text{s.t.} & W\,y & = h(\xi) - T(\xi)\,x & \\ & & y & \geq 0. & \end{array} \tag{2.2}$$

The recourse problem (2.2) is to determine a recourse decision y ($y \geq 0$, $y \in \Re^{\overline{n}}$) that satisfies the second stage constraints $Wy = h(\xi) - T(\xi)x$ and minimizes the second stage cost $q(\eta)y$ after x is decided and the value of $(\xi,\eta) = \omega$ is known. In (2.2), randomness is allowed to occur in the second stage cost vector $q(\eta) \in \Re^{\overline{n}}$, in the second stage right-hand vector $h(\xi) \in \Re^{\overline{m}}$, and in the technology matrix $T(\xi)$ which is of dimension $\overline{m} \times \overline{n}$. The random components ξ, where $\xi \in \Xi \subset \Re^K$, represent the randomness in the second stage right-hand side and the technology matrix, and the random components η, where $\eta \in \Theta \subset \Re^L$, represent the randomness in the second stage cost vector. The vectors ξ and η together constitute the joint random vector ω. The $\overline{m} \times \overline{n}$ recourse matrix W is

deterministic. In a more general setting, the recourse matrix W may be allowed to be random too, such as in some financial applications. But we assume that W is a fixed matrix. Hence, the model SLP_2 in (2.1)-(2.2) is termed a stochastic linear program with *fixed recourse.*

We assume that the $(K+L)$-dimensional random vector $\omega = (\xi, \eta)$ is defined on a probability space $(\Omega, \mathcal{A}, P)$ which is assumed to be determined independently of x. Ω is the domain of the joint random vector ω. We also assume that $q(\eta)$, $h(\xi)$, and $T(\xi)$ can be represented by the following affine functions:

$$q(\eta) = q_0 + \sum_{l=1}^{L} q_l \eta_l, \quad h(\xi) = h_0 + \sum_{k=1}^{K} h_k \xi_k\,, \quad T(\xi) = T_0 + \sum_{k=1}^{K} T_k \xi_k\,, \tag{2.3}$$

where $h_0, h_1, \ldots h_K \in \Re^{\overline{m}}$ and $q_0, q_1, \ldots q_L \in \Re^{\overline{n}}$ are fixed vectors, and $T_0, T_1, \ldots T_K \in \Re^{\overline{m} \times \overline{n}}$ are fixed matrices. The assumption is justified since one can always identify basic random components such that (q, h, T) are linear functions of those basic random components. The motivation of using representation in (2.3) is to reduce the dimension of uncertainty in (q, h, T) to that of ω (see Kall [44]). Components of $\omega = (\xi, \eta)$ may be dependent.

2.1.2 Model properties

In solving SLP_2 (2.1)-(2.2), the selection of a feasible first-stage decison x is affected implicitly by the second stage problem in two ways. First, one needs to limit to the set of acceptable first stage decisions which will result in a feasible second-stage problem in (2.2). Secondly, for each selected x, one must take into account the expected cost of the second stage decision such an x may generate. To ensure feasibility of SLP_2, the set of recourse decisions is assumed to be *complete* in the sense that, whatever the choice of x there is a

possible choice of y to correct the discrepancy between $T(\xi)x$ and $h(\xi)$. $T(\xi)x$ is customarily interpreted as a *tender* to be bid against future uncertain outcomes $h(\xi)$.

To define the feasibility of SLP_2 formally in terms of the first stage decision x, in accordance with Wets [71], let X^1 denote the feasible set determined by the first stage (deterministic) constraints, i.e.,

$$X^1 := \{x \in \Re^n \mid Ax = b,\ x \geq 0\}, \tag{2.4}$$

and let X^2 denote the feasible set induced by the feasibility of the second stage recourse problem, known as *induced feasible set*:

$$X^2 := \{x \in \Re^n \mid \exists\, y \geq 0 \text{ s.t. } Wy = h(\xi) - T(\xi)\,x,\ \xi \in \Xi\}. \tag{2.5}$$

Then X, the set of acceptable (feasible) first stage decisions, is the intersection of X^1 and X^2, i.e.,

$$X := X^1 \cap X^2. \tag{2.6}$$

SLP_2 is said to be *feasible* if $X \neq \emptyset$. If $X^1 \subseteq X^2$, SLP_2 is said to have *relative complete recourse* which means that for any x feasible in the first stage, the second stage problem is always feaible for any realization of the random vector ω. Furthermore, if $X^2 = \Re^n$, SLP_2 is said to have *complete recourse* which means that the feasibility of the second stage recourse problem is ensured always.

Assuming complete (or at least relatively complete) recourse is not restrictive from the standpoint of applications, since in practice, almost always there is a feasible recourse action. Even if there is not, one can model the second stage recourse problem with high penalty for constraint violation. Consequently, W is assumed to be a complete recourse matrix.

A special case of complete recourse is when $W = (I, -I)$ which is known as *simple recourse* because the recourse $y = (y^+, y^-)$ in (2.2) is trivially determined once the decision x is made and the random vector ω is realized. Stochastic programs with simple recourse essentially models the allocation of scarce resources under uncertainty with penalties associated with surplus (y^+) and shortage (y^-). The recourse $y = (y^+, y^-)$ is not really an action but more a "record of state". With any decision x, the simple recourse problem is always feasible. Define the set of feasible recourse decisions by:

$$Y(x, \xi) := \{y \mid Wy = h(\xi) - T(\xi)x,\ y \geq 0\}. \tag{2.7}$$

The property of simple recourse implies that the set Y has precisely one element for each (x, ξ), and thus, the second-stage optimization is fictitious and can be explicitly determined. Based on the special properties of simple recouse, extensive theoretical results and efficient solution procedures have been developed for the case of stochastic programs with simple recourse. See, for instance, Wets [73] and Ziemba [76], for details.

Since we are dealing with a general class of stochastic linear programs with fixed recourse, in what follows, properties that are frequently used in developing solution methods for stochastic linear programs with fixed recourse are stated without proof.

(i) Each of the feasible sets X^1, X^2, X, and Y is convex.

(ii) The recourse function $\phi(x, \omega)$ is a convex (piecewise linear) function in x for each $\omega = (\xi, \eta) \in \Omega$.

(iii) The recourse function $\phi(x, \xi, \eta)$ is a convex-concave (piecewise linear) saddle function in (ξ, η) for each $x \in X$. If $L = 0$ (no uncertainty in the objective function coefficients),

ϕ is reduced to a convex (piecewise linear) function in ξ for each $x \in X$. Similarly, if $K = 0$ (uncertainty in the right-hand side is absent), ϕ is a concave (piecewise linear) function in η for each $x \in X$.

(iv) In addition to the assumption of complete recourse, dual feasibility of the second stage recourse problem (2.2):

$$D := \left\{ \pi \in \Re^{\overline{m}} \mid \pi W \leq q(\eta),\ \eta \in \Theta \right\} \neq \emptyset, \tag{2.8}$$

ensures that ϕ is a *proper* function. Since the case of unboundedness is of no interest to us, we shall also assume that the above condition is satisfied for each realization of the random vector ω.

For detailed discussion concerning the properties of two-stage stochastic linear programs, consult Kall [44] and Wets [71].

2.1.3 Case of discrete distributions

Here, we consider the case that the underlying distribution of ω is discrete or is discretized. The random vector ω takes on only finite number of possible values $\omega^s = (\xi^s, \eta^s)$ with associated probabilities p_s, $p_s > 0$, $s \in S$, where S is an index-set and $\sum_{s \in S} p_s = 1$. Each possible value ω^s, $s \in S$ corresponds to a *scenario* of the occurrence of the random events. In order to better describe the uncertainty faced in a real-world stochastic decision problem, the scenario set S may be chosen large in size, i.e., $|S|$, the cardinality of the set S, is large.

In this case, SLP_2 (2.1)-(2.2) takes on the form

$$\begin{array}{lll} \min_x & \{\ c\,x & + \sum_{s \in S} p_s\, \phi(x, \omega^s)\ \} \\ \text{s.t.} & A\,x & = b \\ & x & \geq 0, \end{array} \tag{2.9}$$

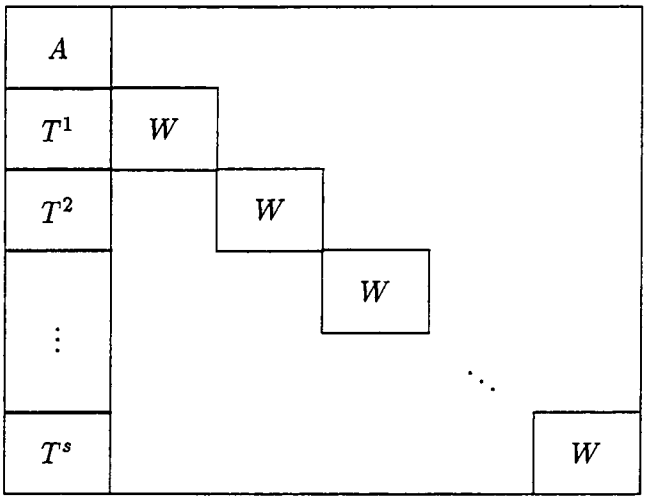


Figure 2.1: Structure of a discrete 2-stage stochastic program.

where the recourse function $\phi(x,\omega^s)$ under each scenario ω^s, $s \in S$, is determined by

$$\begin{array}{lllll} \phi(x,\omega^s) := & \min_y & q(\eta^s)\, y & & \\ & \text{s.t.} & W\, y & = & h(\xi^s) - T(\xi^s)\, x \\ & & y & \geq & 0. \end{array} \tag{2.10}$$

The problem (2.9)-(2.10) is equivalent to determining x and y^s, $s \in S$ for the following large scale deterministic linear program, termed *Grand* LP for two-stage models (GLP_2):

$$\begin{array}{llll} (\text{GLP}_2) \quad \min_{x,y^s} & \{\, cx \,+\, \sum_{s\in S}\, p_s\, q(\eta^s)\, y^s\} & & \\ \text{s.t.} & A\, x & = & b \\ & T(\xi^s)\, x \,+\, W y^s & = & h(\xi^s) \quad s \in S \\ & x \geq 0,\ y^s \geq 0 & & s \in S. \end{array} \tag{2.11}$$

Observe that GLP_2 has a *dual block angular* structure, depicted in Figure 2.1, which can be exploited for the application of mathematical decomposition techniques. For example, the dual of GLP_2 is in the standard form for applying Dantzig-Wolfe decompostion [13], or one can apply an L-shaped (Bender's) decomposition [70] to the problem of the primal

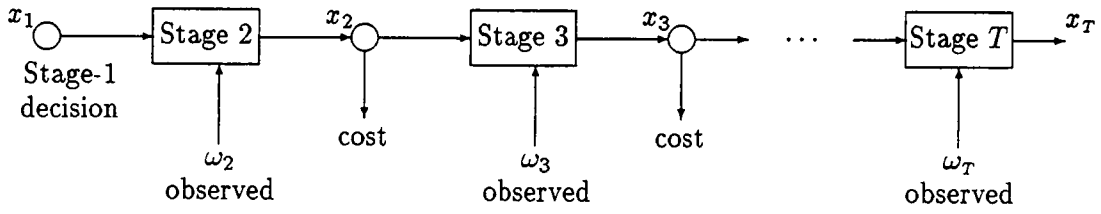


Figure 2.2: T-stage stochastic decision sequence.

form. For a detailed discussion about large scale linear programming techniques, refer to Wets [75].

2.2 Multi-stage models

A multi-stage stochastic decision model is similar to the two-stage problem except that future recourse decisions are made sequentially in many stages over the planning time horizon, depending on how information about uncertainty is revealed and gathered.

2.2.1 Model formulations

Consider a T-stage stochastic decision problem, where $T \geq 2$. After the first stage decision x_1 is made prior to the knowledge of the outcome of random event $(\omega_2,\ \omega_3 \ldots \omega_T)$, in each stage t, $t = 2, \ldots T$, the outcome of random events ω_t occurring at stage t is realized and, correspondingly, a recourse decision x_t is then determined. The decision-making and information gathering sequence is depicted in Figure 2.2.

The information about stochastic outcomes $\omega = (\omega_2,\ \omega_3, \ldots, \omega_T)$ is revealed partially in each decision stage. Decisions x_t made in stage t, $t = 1, \ldots T$, can depend only on the decisions made in the previous stages $(x_1, \ldots, x_{t-1})$, and on the outcomes realized up to stage t, $(\omega_2, \ldots, \omega_t)$, but not on the hindsight of future outcomes $\omega_{t+1}, \ldots, \omega_T$. This require-

ment of the decisions $x = (x_1, \ldots, x_T)$ is known as *nonanticipativity*. The nonanticipativity constraints on x require that the same decision sequence $(x_1, \ldots, x_t)$ would be taken for all scenarios that share the same history of outcomes $(\omega_2, \ldots \omega_t)$ up until period t. In other words, the decision sequence must be adapted to the information filtering described by the sequential revelation of ω_t.

A formulation of this decision problem is the following minimum cost T-stage stochastic linear program (SLP_T):

$$\begin{aligned} \min_{x_1} \quad & \{ \; c_1 x_1 + E_{\omega_2}[\min_{x_2} c_2(\eta_2) x_2 \; + \; E_{\omega_3|\omega_2}[\min_{x_3} c_3(\eta_3) x_3 + \ldots \\ & \qquad \ldots \; + \; E_{\omega_T|(\omega_2, \ldots \omega_{T-1})}[\min_{x_T} c_T(\eta_T) x_T] \ldots]] \; \} \\ \text{s.t.} \quad & A_1 x_1 = b_1 \\ & B_t(\xi_t) x_{t-1} + A_t x_t = b_t(\xi_t) \;\; t = 2, \ldots T \\ & x_t \geq 0, \quad t = 1, \ldots T \\ & x \text{ nonanticipative i.e., } \; x_t \equiv x_t(x_1, \ldots x_{t-1}; \omega_2, \ldots \omega_t). \end{aligned} \tag{2.12}$$

This is a fixed recourse model, since the recourse matrices A_t, $t = 2, \ldots T$, are deterministic.

The stochastic outcomes realized in stage t, $t = 2, \ldots T$ are $\omega_t = (\xi_t, \eta_t)$, where $\eta_t \in \Re^{L_t}$ represents the randomness in the cost vector c_t and $\xi_t \in \Re^{K_t}$ represents the randomness in the right-hand side vector b_t and in the technology matrices B_t. The joint random vector $\omega = (\omega_2, \; \omega_3 \ldots \omega_T)$ is assumed to be defined on a probability space $(\Omega, \mathcal{A}, P)$. Dependence of random components may exist, and the type of dependence may be within time periods (intra-period dependence) as well as between time periods (inter-period dependence). The operator $E_{\omega_t|(\omega_2, \ldots \omega_{t-1})}$ denotes conditional expectation with respect to ω_t given $(\omega_2, \ldots \omega_{t-1})$, $t = 3, \ldots T$. The random vectors c_t and b_t and the random technology matrices B_t, $t = 2, \ldots T$, are assumed to be affine functions in the corresponding random vectors ξ_t and η_t. The dimensions of arrays present in the model are as follows: $x_t \in \Re^{n_t}$,

$c_t \in \Re^{n_t}$, $b_t \in \Re^{m_t}$, A_t of dimension $m_t \times n_t$, and B_t of dimension $m_t \times n_{t-1}$, $t = 1, \ldots T$.

The T-stage model SLP_T can be viewed as a nested sequence of two-stage recourse models in which the recourse function at stage t, $\phi_t(x_1, \ldots, x_{t-1}, \omega_2, \ldots, \omega_t)$, $t = 2, \ldots T-1$, can be defined recursively as follows:

$$\begin{array}{lllll} \phi_t \equiv & \min_{x_t} & \{\ c_t(\eta_t)\, x_t & + E_{\omega_{t+1}|(\omega_2,\ldots,\omega_t)}\, \phi_{t+1}(x_1, \ldots, x_t, \omega_2, \ldots, \omega_{t+1})\ \} \\ & \text{s.t.} & A_t\, x_t & = b_t(\xi_t) - B_t(\xi_t)\, x_{t-1} \\ & & x_t & \geq 0, \end{array} \tag{2.13}$$

where the recourse function at the terminal stage T, $\phi_T(x_1, \ldots, x_{T-1}, \omega_2, \ldots, \omega_T)$, is:

$$\begin{array}{lllll} \phi_T \equiv & \min_{x_T} & c_T(\eta_T)\, x_T & & \\ & \text{s.t.} & A_T\, x_T & = b_T(\xi_T) - B_T(\xi_T)\, x_{T-1} \\ & & x_T & \geq 0, \end{array} \tag{2.14}$$

From the above formulation, it is apparent that decisions x_t made in stage t, $t = 2, \ldots T-1$, are not just recourses to discrepancies generated by the decisions made in the previous stages, they are also risk-taking decisions which take into account future uncertainty. The terminal stage decision x_T is the only (recourse) decision which is made with full information about uncertain outcomes.

Again, we assume, without loss of generality, that the model has complete recourse, or in other words, there are no such feasible sets of values of B_τ and $x_{\tau-1}$, $\tau < t$, $1 < t \leq T$, that make the problem infeasible with respect to the decision x_t.

Many theoretical results for two-stage models have been extended to T-stage models. For results regarding measure-theoretic issues, characteristics of objective functions, conditions under which the strong duality holds, see Olsen [59] [60] [61] and Rockafellar and Wets [66].

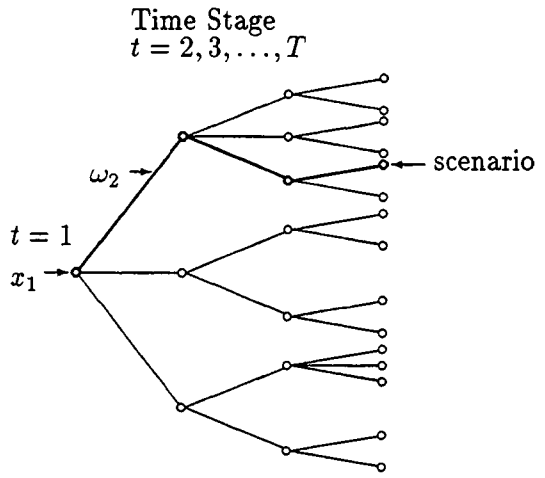


Figure 2.3: A scenario tree.

2.2.2 Case of discrete distributions

Now consider the case that the random vector ω takes on a finite number of possible scenarios $\{\omega^s : s \in S\}$ with associated probabilities $\{p_s: s \in S\}$, where S is a finite set with presumably large $|S|$.

Let x_t^s denote the decisions made under scenario ω^s at stage t, $t = 2, \ldots T$. By nonanticipativity, if two different scenarios ω^s and $\omega^{s'}$ are indistinguishable at time τ on the basis of information available about them at time τ, then $x_t^s = x_t^{s'}$ for $t = 1, \ldots \tau$. To model these crucial constraints, the decision information structure generated by $\{\omega^s : s \in S\}$ is illustrated by a scenario tree structure in Figure 2.3.

In a scenario tree, the root node represents the first stage decision, each of the other branching node represents an opportunity to make a (recourse) decision, and each branch represents a possible outcome of the uncertain events occurring in that stage. Each path through the tree is a scenario representing one possible sequence of outcomes of the random

events throughout the time horizon considered in the problem. Different scenarios may share a common path up to some stage and then diverge. Those scenarios passing through the same node at stage t generate the same information up to stage t and should generate the same sequence of decisions up to stage t. Let $\mathcal{N}_t$, $t = 2, \ldots T$, denote the set of branching nodes at stage t and let N denote any particular node in the scenario tree. We denote the set of scenarios ω^s passing through a particular node N at stage t, by $\{s : s \in S, s \in N\}$ where $N \in \mathcal{N}_t$. Nonanticipativity constraints impose the requirement that x_t^s must be constant for those scenarios ω^s containing the same information up to stage t. The nonanticipativity constraints can be expressed explicitly in terms of the information structure as

$$x_t^s = z_t^N \ \forall s \in N, s \in S, \ N \in \mathcal{N}_t, \ t = 2, \ldots T-1, \tag{2.15}$$

where z_t^N is a decision made based on the information revealed at the branching node N. The problem SLP_T in (2.12) can be reformulated equivalently as the following large scale linear program, Grand LP for T-stage models (GLP_T):

$$\begin{aligned} (\text{GLP}_T) \quad \min_x \quad & \left\{ c_1 x_1 \ + \ \textstyle\sum_{t=2}^{T} \sum_{s \in S} \ p_s \ c_t(\eta^s) \ x_t^s \right\} \\ \text{s.t.} \quad & A_1 x_1 = \ b_1 \\ & B_t(\xi_t^s) \, x_{t-1}^s \ + \ A_t x_t^s = \ b_t(\xi_t^s) \ \ s \in S, \ t = 2, \ldots T \\ & x_t^s \geq \ 0 \ \ s \in S, \ t = 2, \ldots T \\ & x_t^s = z_t^N \ \ \forall s \in N, \ s \in S, \ N \in \mathcal{N}_t, \ t = 2, \ldots T-1, \end{aligned} \tag{2.16}$$

The above formulation is the so-called scenario formulation of the T stage problem, and it possesses a special block structure in the (grand) constraint matrix.

Chapter 3

Existing Solution Methods

In this chapter, an overview of the existing solution methods for solving stochastic (linear) programs is presented. The existing solution methods can be classified broadly into the following three categories: mathematical decomposition-based approaches, sampling-based approaches, and bounds-based approximation approaches. In the following, we outline the main characteristics of each approach and overview some of the prominent methods.

3.1 Mathematical Decomposition-based Approaches

These approaches exploit the special structure in GLP_2 or GLP_T. The main characteristic of these approaches is to decompose a large problem into smaller problems and connect these problems in some computationally-efficient manner.

3.1.1 L-shaped Decomposition

This method is essentially a pure LP version of Benders decomposition [4], and can be viewed as a cutting hyperplane algorithm (outer linearization). Development of this method for two-stage stochastic programs is discussed in Van Slyke and Wets [70].

Consider the problem GLP_2 in (2.11) which is assumed feasible and bounded. This

method first solves the following master problem (MP) :

$$\begin{array}{lll} \min_{x,\theta} & cx + \theta & \\ \text{s.t.} & Ax & = b \\ & x \geq 0, & \end{array} \tag{3.1}$$

where the variable θ is lower estimate of the expected recourse function. For a given x, subproblems $\phi(x, \omega^s)$, $s \in S$, as defined in (2.10), are solved. If $\phi(x, \omega^s)$ is infeasible for some s, $s \in S$, a constraint of the form $Dx \geq g$ is added to the master problem MP in (3.1). The added constraint (called a feasibility cut) is determined, for a specific s, by

$$D = \sigma^s T(\xi^s), \quad g = \sigma^s h(\xi^s), \tag{3.2}$$

where σ^s is an optimal dual solution of the Phase I problem of $\phi(x, \omega^s)$. After the constraint is added, the (restricted) MP is solved again. The process continues until a solution x is found such that all the subproblems $\phi(x, \omega^s)$, $s \in S$, are feasible. In this case (since a feasible solution is available), the expected recourse value needs to be evaluated and compared with the estimate θ. If θ is also an upper for the expected recourse function, i.e., $\sum_{s \in S} p_s \phi(x, \omega^s) \leq \theta$, then the solution x obtained is an optimal solution. Otherwise, a constraint (optimality cut) of the form $\widehat{D}x + \theta \geq \widehat{g}$ is added to the (restricted) MP. The added optimality cut is determined by

$$\widehat{D} = \sum_{s \in S} p_s \pi^s T(\xi^s), \quad \widehat{g} = \sum_{s \in S} p_s \pi^s h(\xi^s), \tag{3.3}$$

where π^s is an optimal dual solution of subproblem $\phi(x, \omega^s)$, $s \in S$. The process continues to find a feasible solution x that generates as least cost as possible. This procedure is finite and convergent.

When applying the above procedure to solve multi-stage problems, one can solve each of second stage subproblems $\phi_2(x_1, \omega_2^s)$, $s \in S$, as a large linear program, or consider

next stage subproblems of each second stage subprolem. The latter approach is known as *nested* decomposition. This method is described in Birge [5], and its implementation can be found in Gassmann [37]. For an application of this method to quardratic problems, see Louveaux [52].

3.1.2 Progressive Hedging Algorithm

The Progressive Hedging Algorithm (PHA), also known as the Scenario Aggregation (SA) algorithm, is due to Rockafellar and Wets [68].

The idea is to determine the optimal decision policy for each scenario, a procedure known as *scenario analysis*, and consequently find one decision that hedges well against all possible scenarios by combining scenario decisions relative to their probabilities.

For each scenario ω^s, $s \in S$, consider a scenario subproblem (ignoring nonanticipativity requirements)

$$\begin{array}{lll} \min_{x^s} & \{\, c_1 x_1^s + \sum_{t=2}^{T} c_t(\eta_t^s) x_t^s \,\} & \\ \text{s.t.} & A_1 x_1^s & = b_1 \\ & B_t(\xi_t^s) x_{t-1}^s + A_t x_t^s & = b_t(\xi_t^s) \quad t = 2, \ldots T \\ & x^s \geq 0. & \end{array} \tag{3.4}$$

A solution x^s to the scenario subproblem is known to be *admissible* for scenario ω^s, but is not necessarily *implementable* when all possible scenarios are considered since the nonanticipativity requirements may not be satisfied. The solution procedure first obtains an approximate implementable solution $\bar{x}$ by aggregating all scenario solutions x^s according to their probabilities. Each scenario subproblem (3.4) is then solved again with its objective function replaced by an augmented Lagrangian objective:

$$\min_{x^s} \left\{ c_1 x_1^s + \sum_{t=2}^{T} c_t(\eta_t^s) x_t^s + \alpha_s x^s + \frac{r}{2} \parallel x^s - \bar{x} \parallel^2 \right\}, \tag{3.5}$$

which dualizes the nonanticipativity constraints and imposes a quadratic penalty on deviations from $\bar{x}$. Penalty parameter is denoted by $r > 0$ and α_s is an estimate of the dual multipliers associated with the nonanticipativity constraints $x^s - x = 0$, see (2.16). Obtain a new approximate implementable decision $\bar{x}$ by aggregating new scenario decisions x^s, $s \in S$. At this point, check convergence of approximate solution $\bar{x}$. One of the termination criteria used in Rockafellar and Wets [68] is

$$\delta = \| \bar{x} - \bar{x}^{\nu+1} \|^2 + \sum_{s \in S} p_s \| x^s - \bar{x} \|^2, \tag{3.6}$$

which monitors both the primal solution x and the dual prices α for convergence. The solution procedure is repeated until the termination criterion is satisfied. This algorithm is convergent from any starting point $(\bar{x}^0, \alpha^0)$.

Extensions of the scenario aggregation technique for problems with nonseparable convex constraints (in addition to the nonanticipativity constraints) have been derived in Robinson [63]. The computational behavior of the progressive hedging algorithm has been studied in Mulvey and Vladimirou [56, 57, 58]

3.2 Sampling-based Approaches

The basic idea is to draw random samples iteratively from the underlying probability distribution to compute sample-based estimates of subgradients or cutting planes for instance, in the process of approximating expected values. Such solution algorithms are developed based upon asymptotic convergence theory and statistical termination criteria.

3.2.1 Stochastic Decomposition

This method was proposed by Higle and Sen [39]. Analogous to the L-shaped decomposition, this method involves solving master problems and subproblems, and iteratively generating a sequence of piecewise linear objective function approximations. However, instead of solving subproblems for all $\omega^s \in \Omega$ to create a cut as in L-shaped decomposition, the stochastic decomposition (SD) procedure, in each iteration ν, solves only one subproblem for a randomly sampled point ω^ν from the given distribution and $\nu - 1$ strongly relaxed recourse problems. The problems considered by this method are two-stage models with complete recourse, nonstochastic q and T (also applicable for stochastic T), and $\phi(x, \omega) > 0$.

3.2.2 Decomposition with Importance Sampling

The main idea is to reduce the variance of statistical estimates within sample-based L-shaped decomposition procedures by using variance reduction techniques such as *importance sampling.*

In each iteration of the L-shaped decomposition, instead of solving subproblems for all $\omega^s \in \Omega$, this method samples a subset of ω^s, $s \in S^\nu \subset S$ for which subproblems will be solved in iteration ν. An estimate of the expected recourse cost $E[\phi(x, \omega^s)]$ is computed, based on the sampled scenarios ω^s, $s \in S^\nu$. From the central limit theorem, it is known that the error of the estimate using Monte Carlo (random) sampling is approximately normally distributed with standard deviation inversely proportional to $\sqrt{|S^\nu|}$. The smaller the standard error is the better the estimate is. However, increasing the sample size $|S^\nu|$ can only reduce the standard error slightly.

This method applies importance sampling to reduce the standard error effectively. The

idea behind importance sampling is to change the sampling procedure so that those events that matter more (have significant contribution to the expected recourse cost) are sampled more. Those scenarios having a major impact on the solution's ability to hedge against uncertainty typically have low probabilities, and hence will rarely be picked up by a random sampling procedure. By sampling the random events according to their importance, the solution is forced to hedge properly. For details, see Dantzig and Glynn [14].

3.2.3 Stochastic Quasi-Gradient Methods

The stochastic quasi-gradient (SQG) technique has been introduced by Ermoliev [26, 27], specially for dealing with stochastic programs involving continuous random variables. This method constructs a sequence of approximate solutions x^ν, $\nu = 0, 1, \ldots$, by using sample-based estimates of subgradients, also called stochastic quasi-gradients. The general procedure of the SQG method at iteration ν with x^ν given, involves sampling a point ω^ν from the given distribution, calculating a sample-based estimate of the subgradient at x^ν, constructing an approximate solution $x^{\nu+1}$ by projecting x^ν onto the feasible set X using x^ν and a prescribed step size. The convergence of solution sequence $\{x^\nu\}$ depends greatly on the choice of step sizes.

For further details of this and other variants of this technique, see Ermoliev [26, 27].

3.3 Bounds-based Approximation Approaches

The main idea behind these approaches is to replace the original problem with approximate problems such that approximate solutions and their error estimates can be easily computed. The general solution procedure used by these approaches is the sequential approximation method (SAM) which requires approximations of the original problem, error estimates to

evaluate approximations, and refining schemes so that convergent solution sequences can be constructed.

When considering the two-stage problem SLP_2 in (2.1), one focuses on approximating $E_\omega\phi(x,\omega)$. The aim usually is to obtain lower and upper bounding approximate functions, $\psi_L(x)$ and $\psi_U(x)$ respectively, such that

$$\psi_L(x) \leq \psi(x) := E_\omega\phi(x,\omega) \leq \psi_U(x), \tag{3.7}$$

and that evaluating $\psi_L(x)$ and $\psi_U(x)$ is relatively easy compared to evaluating $E_\omega\phi(x,\omega)$. Usually error estimates are provided by these bounding approximate functions, therefore, one would desire to have the bounding approximate functions in (3.7) to bracket $E_\omega\phi(x,\omega)$ as tightly as possible, and in particular, when ϕ is linear (or multi-linear) in ω, these bounding approximate functions should coincide with $\psi(x)$.

3.3.1 Approximation Schemes

Construction of approximations exploits mainly the properties of ϕ (such as convexity, concavity, or convex-concave saddle property) and certain moment information of the underlying distribution, such as the first moments, cross moments or other higher order moments.

Bounding functions usually result from approximating the recourse function $\phi(x,\omega)$ or from approximating the distribution of ω. Generally, approximations for the recourse function use linear or bilinear approximate functions, and approximations of the underlying distribution are discrete measures involving only a few number of points.

The general approach for obtaining bounds on the expected value of the two-stage recourse function is through approximating the underlying distribution of ω by distribution approximations $\{ (\widehat{\omega}^i,\ \widehat{p}^i) : i = 1,\ldots I \}$, where each $\widehat{\omega}^i$ is associated with probability

$\widehat{p}^i$, $i = 1, \ldots I$. Various approximations have been derived depending on the following conditions: the geometric shape of the uncertainty domain Ω (or Ξ, Θ), moment information of random variables, independence (or dependence) assumption on random components, and the properties of the recourse function. As an example, if the recourse function is convex with univariate random variable ξ distributed in a closed interval $[a, b]$, with only first moment $\overline{\xi}$ available, Edmundson-Madansky (E-M) inequality [54, 55]:

$$E\phi(x, \xi) \leq \frac{b - \overline{\xi}}{b - a} \phi(x, a) + \frac{\overline{\xi} - a}{b - a} \phi(x, b), \tag{3.8}$$

provides an upper bound on the expected value of the recourse function, and $\{(a, \lambda_1), (b, \lambda_2)\}$, where $\lambda_1 = \frac{b-\overline{\xi}}{b-a}$ and $\lambda_2 = \frac{\overline{\xi}-a}{b-a}$, is an approximation of the underlying distribution on the interval $[a, b]$. With independence assumption on random variables, E-M upper bound can be extended straightforwardly for multivariate convex case.

For convex cases, the studies on approximations have been extensive. Using first moment information without independence assumption, Gassmann and Ziemba [38] and Birge and Wets [8] derived upper bounds for unbounded domains. With first and joint moments and with compact rectangular domains, upper bounds were derived for independent and dependent cases in Frauendorfer and Kall [36] and Frauendorfer [30], respectively. Without independence assumptions, Dulá [17] and Kall [47] developed upper bounds using first moments and total second moment, and Dulá and Murthy [18] generalized their previous work by using first and marginal second moments. Most of the approximations mentioned above solve their related GMP. Refer to Kall [46] for a review, For more general cases, the cases of convex-concave saddle recourse function, the studies of approximation are relatively rare. Without any independence assumption, Frauendorfer [31] derived approximations of P

using first and joint moments for compact rectangular domains $\Omega = \Xi \times \Theta$, Frauendorfer [33] derived approximations requiring first and cross moments for compact simplicial marginal domains Ξ and Θ, Edirisinghe and Ziemba [23, 24] developed approximations using first and cross moments for joint domains $\Omega \supseteq \Xi \times \Theta$ being arbitrary polytopes and for unbounded joint domains, respectively, and Edirisinghe [19] derived approximations using first, cross, and second moments for simplicial joint domains and approximations using first, joint, and second moments for rectangular joint domains.

The type of approximations used affects computational effort required for evaluating approximate functions by two ways: by the sizes of the bounding problems and by the tightness of the resultant bounds. The sizes of the bounding approximate problems usually determined by the number of points in the approximate distributions, which, in turn, is determined by the number of extreme points of the domains used for approximation. For example, in the cases of saddle recourse functions with $K + L$ variables, approximations using rectangular marginal or joint domains result in problem size of 2^{K+L}, those using simplicial marginal domains Ξ and Θ result in problem size of $(K + 1)(L + 1)$, and those using simplicial joint domain enclosing $\Xi \times \Theta$ result in problem size of $(K+L+1)$. Therefore, for reducing computational effort due to the sizes of the bounding problems, approximations using simplicial joint domain should be chosen over the other options.

Regarding the tightness of the resultant bound due to the amount of moment information used in approximation, although it is not clear yet, approximations using more moment information should result in tighter bounds, since more information such as variance and covariance about variables (or uncertainty) is known. If we define *first-order* information of the underlying distribution as those moment information in which each random variable

is raised at most to the power of one, and similarly define *second-order* information as those moment information in which each random variable is raised at most to the power of two, then it is obvious that second-order moment information reveal more about random variables than first-order moment information.

3.3.2 Refinement Schemes

The basic technique for refining approximations and improving bounds is to subdivide the given domains into smaller *cells* and reapply the approximation procedure using conditional moment information with respect to the new partition. Subdividing a cell is done by intersecting hyperplanes to coordinate axes of the uncertainty domain. For a rectangular cell, the simplest way to subdivide it into rectangular sub-cells is to partition the cell by a hyperlane orthogonal to coordinate axes.

As more sub-cells are created, additional computational effort is required to compute the conditional moment information of all cells and to solve larger bounding problems. Consider the bounding problems constructed through probability approximation on the extreme points of a rectangular domain $\Omega \subseteq \Re^{K+L}$. With an additional cell created, the size of a bounding problem increases by 2^{K+L} blocks, an exponential increase of problem size with the dimension of random variables. Therefore, effective partitioning strategies are needed so that the bounds can be improved most with only a few refining steps.

In designing partitioning strategies to refine approximations and improve bound, it is important to know that the tightness of the bounds depends crucially upon how close $\phi(x, \cdot)$ is being linear. All the function approximations derived so far are exact if $\phi(x, \cdot)$ is linear (or multilinear) in ω. Subdividing a cell in which $\phi(x, \cdot)$ is linear will not improve the

approximations. Thus, one strategy for efficiently improving approximations is to choose those cells that capture as much of the nonlinearity of $\phi(x,\cdot)$ as possible for refinement.

For the purpose of effectively refining approximations, a partitioning strategy determines which cells to be subdivided and how these cells should be subdivided with regard to the nonlinearity of $\phi(x,\cdot)$. A cell should be selected for partition if the difference between upper and lower bounds exceed the prescribed tolerance, because if $\phi(x,\cdot)$ is linear in a cell then there is no approximation error, i.e., $\psi_L(\overline{x}) = \psi_U(\overline{x})$. Note that for partitioning purpose the lower and upper bounds should be compared at the same x solution, as $\phi(x,\cdot)$ depends on x. When a cell is chosen to be subdivided, a coordinate (or an edge) need to be selected to be intersected with the partitioning hyperplane. Again, we should select a direction along which the function $\phi(x,\cdot)$ is most nonlinear for splitting. Computing measures of nonlinearity of $\phi(x,\cdot)$ in a rectangle involves dual analyses. For details, refer to Frauendorfer and Kall [36], Frauendorfer [30] and Edirisinghe and Ziemba [25]. The last step in subdiving a cell, is to select a splitting point which usually is a midpoint or a mean point.

A cell-redefining strategy was introduced and implemented in Edirisinghe and Ziemba [25] to further reduce the computational effort required in refining steps. Cell-redefining procedure relocates extreme points that define a new sub-cell as compact as possible after subdividing a cell containing discrete scenarios. The rationale is that better approximations should result in tighter bounds which consequently is expected to result in a smaller number of refining steps to achieve the prescribed accuracy.

3.3.3 Convergence

Generally, if the domain of the random variables is bounded and if the probability of each cell goes to zero as the number of partitions goes to infinity, the bounds converge to the true function values. Regarding to the convergence of the sequence of approximate x solutions, some additional conditions, for example, compactness of the feasible set for x together with the uniform convergence of bounds and continuity of the objective function, are needed to ensure such convergence. The convergence theory for solving SLP_2 with bounds-based approximation approach is built on the concept of *epi-convergence*. Relevant theories and details regarding epi-convergence can be found in Attouch and Wets [1], Wets [72], and Birge and Wets [8].

3.4 Research scope

This research deals with the development, design, and implementation of new bounds-based approximation methodology for solving multi-period stochastic programming recourse models. The specific objectives of this research are:

(i) development of refining procedures for the second-order simplicial approximations and their implementation for solving two-stage models.

(ii) computational experiments and validation of second-order approximations solution method on a large set of two-stage test problems.

(iii) development of new approximations for multi-stage models and design solution procedures for solving multi-stage models.

(iv) investigate implementational aspects and design decomposition-based techniques for efficient computation of the new approximations.

(v) Perform a detailed computational analysis of the developed techniques.

Chapter 4

Solution Method for Two-stage Models

In this chapter, a solution method based on second-order simplicial approximations [19] is developed and implemented for solving two-stage stochastic linear program GLP_2 given in (2.11) in which uncertainty is described by the set of scenarios $\{\omega^s := (\xi^s, \eta^s) \ : \ s \in S\}$ for the $(K + L)$-dimensional random parameters $\omega = (\xi, \eta)$. First, a brief review of the basic second-order simplicial approximation procedure proposed by Edirisinghe [19] is provided. The remainder of the chapter is mostly concerned with developing the theoretical and computational procedures for efficient implementation of a sequential bound refinement methodology. Computational analysis using various heuristic partitioning strategies will also be presented.

4.1 Scenario approximations and clustering

The basic idea behind scenario approximations is to approximate the underlying (given) probability distribution by discrete scenario distributions involving only a small number of points. The construction of scenario approximations depend on utilizing certain moment information of the given distribution and using a specific geometric shape for the uncer-

tainty domains. The computational effort required for solving the bounding problems and the tightness of the resultant bounds are dependent on the particular choice of moments and domain shape. In the sequel, a second-order simplicial approximation is chosen for developing a solution method for two-stage stochastic programs.

4.1.1 Second-order approximations

Consider the joint domain Ω to be a $(K+L)$-dimensional simplex, having extreme points $w^i \equiv (u^i, v^i)$, $i = 1, \ldots I$, where $I = K + L + 1$. Define the $I \times I$-dimensional matrix V by

$$V := \begin{bmatrix} w_1^1 & w_1^2 & \cdots & w_1^I \\ w_2^1 & w_2^2 & \ldots & w_2^I \\ \vdots & & & \vdots \\ w_{K+L}^1 & w_{K+L}^2 & \cdots & w_{K+L}^I \\ 1 & 1 & \ldots & 1 \end{bmatrix}. \tag{4.1}$$

Since Ω is a $(K+L)$-dimensional simplex, V is a nonsingular matrix, the inverse of which is denoted by V^{-1}. Define $\rho = (\rho_1, \ldots \rho_I) \in \Re^I$ by:

$$\rho_i := \left[V^{-1}\right]_i \begin{pmatrix} \sum_{s \in S} p_s \omega^s \\ 1 \end{pmatrix}, \tag{4.2}$$

where $[V^{-1}]_i$ denotes the i-th row of V^{-1}. Define the scenario distribution $\mathcal{D}_L$ by

$$\left\{(\tilde{\omega}^i, \rho_i) \ : \ i = 1, \ldots, I\right\} \tag{4.3}$$

and the scenario distribution $\mathcal{D}_U$ by

$$\left\{(\hat{\omega}^i, \rho_i) \ : \ i = 1, \ldots, I\right\}, \tag{4.4}$$

where $\tilde{\omega}^i = (\tilde{\xi}^i, v^i)$, $\hat{\omega}^i = (u^i, \hat{\eta}^i)$, and $\tilde{\xi}^i = (\tilde{\xi}_1^i, \ldots \tilde{\xi}_K^i)$ and $\hat{\eta}^i = (\hat{\eta}_1^i, \ldots \hat{\eta}_L^i)$ are determined as follows:

$$\tilde{\xi}_k^i = \frac{\tilde{\rho}_i^k}{\rho_i}, \quad \tilde{\rho}_i^k := \left[V^{-1}\right]_i \begin{pmatrix} \sum_{s \in S} p_s \xi_k^s \omega^s \\ \sum_{s \in S} p_s \xi_k^s \end{pmatrix}, \ k = 1, \ldots K, \tag{4.5}$$

$$\hat{\eta}_l^i = \frac{\hat{\rho}_i^l}{\rho_i}, \quad \hat{\rho}_i^l := \left[V^{-1}\right]_i \begin{pmatrix} \sum_{s\in S} p_s \eta_l^s \omega^s \\ \sum_{s\in S} p_s \eta_l^s \end{pmatrix}, \; l = 1, \ldots L. \tag{4.6}$$

Observe that the foregoing scenario distributions $\mathcal{D}_L$ and $\mathcal{D}_U$ are based on the second-order moments of ω with respect to the actual (given) probability distribution.

Due to the convex-concave saddle property of the recourse function $\phi(x,\omega)$ in (2.2), the following upper and lower bounding results have been proven by Edirisinghe [19]:

$$\psi(x) := E_\omega \phi(x,\omega) \;\geq\; \psi_L(x) := \sum_{i=1}^{K+L+1} \rho_i \phi(x, \tilde{\omega}^i), \tag{4.7}$$

$$\psi(x) := E_\omega \phi(x,\omega) \;\leq\; \psi_U(x) := \sum_{i=1}^{K+L+1} \rho_i \phi(x, \hat{\omega}^i). \tag{4.8}$$

Since the preceding bounding inequalities are based on the constructed scenario distributions $\mathcal{D}_L$ and $\mathcal{D}_U$, these are known as second-order approximations. The following bounding approximations for Z^*,

$$Z^* := \min_{x\in X} \left\{ < c, x > + \sum_{s\in S} p_s \phi(x, \omega^s) \right\} \tag{4.9}$$

can be shown to hold:

Theorem 4.1 (Edirisinghe [19])

$$Z^* \leq \min_{x\in X} \left\{ < c, x > \quad + \quad \sum_{i=1}^{K+L+1} \rho_i \phi(x, \hat{\omega}^i) \right\} \tag{4.10}$$

and

$$Z^* \geq \min_{x\in X} \left\{ < c, x > \quad + \quad \sum_{i=1}^{K+L+1} \rho_i \phi(x, \tilde{\omega}^i) \right\}. \tag{4.11}$$

For an illustration of the bounds in Theorem 4.1 for $K = L = 1$, see Figure 4.1 where we have denoted $\zeta^i := (\tilde{\xi}^i, \hat{\eta}^i)$ for $i = 1, \ldots, K+L+1$. Also notice that $\sum_{i=1}^{K+L+1} \rho_i \zeta^i = \bar{\omega} := \sum_{s\in S} p_s \omega^s$.

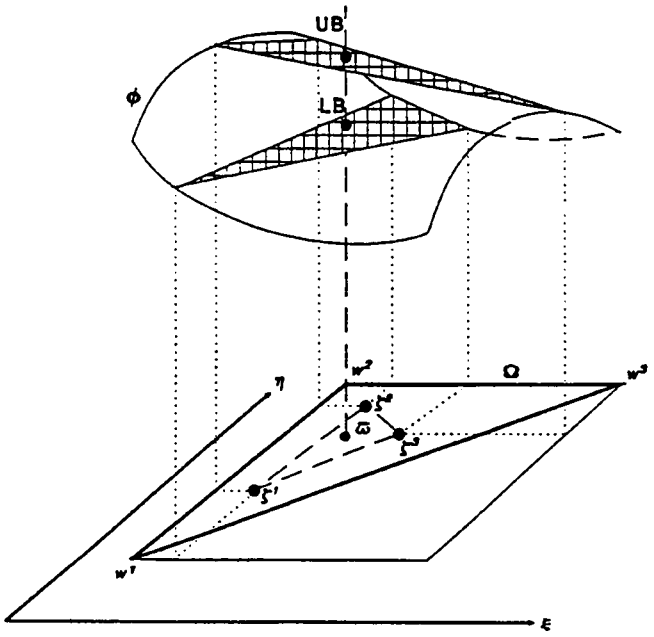


Figure 4.1: Second-moment bounds for the expected recourse function.

The bounding problems in Theorem 4.1 are the following linear programs:

$$\begin{array}{lll} \text{LBLP}: \ \min_{x,y^i} & \left\{cx + \sum_{i=1}^{I} \rho_i \, q(v^i) \, y^i\right\} & \\ \text{s.t.} & Ax & = b \\ & Tx + Wy^i & = h(\tilde{\xi}^i) \quad i = 1, \ldots I \\ & x \geq 0, \ \ y^i \geq 0 \ \ i = 1, \ldots I, & \end{array} \tag{4.12}$$

$$\begin{array}{lll} \text{UBLP}: \ \min_{x,y^i} & \left\{cx + \sum_{i=1}^{I} \rho_i \, q(\hat{\eta}^i) \, y^i\right\} & \\ \text{s.t.} & Ax & = b \\ & Tx + Wy^i & = h(u^i) \quad i = 1, \ldots I \\ & x \geq 0, \ \ y^i \geq 0 \ \ i = 1, \ldots I. & \end{array} \tag{4.13}$$

These LP problems possess the dual angular structure with I blocks. Since $I(= K + L + 1)$ is usually much smaller than $|S|$, the approximating problems can be solved more efficiently.

4.1.2 Partitioning and convergence

For tightening the second-order simplicial approximation, when a partition of the domain Ω is sought, the ensuing cells of the partition are required to be simplices.

Let $\mathcal{S}^\nu := \{\Omega_r \mid r = 1, \ldots, R_\nu\}$ represent such a partition (at some iteration ν) of the joint simplicial domain Ω into subsimplices Ω_r such that

$$\bigcup_{r=1}^{R_\nu} \Omega_r = \Omega, \quad \Omega_r \bigcap \Omega_{r'} = \emptyset, \ \forall r \neq r';\ r, r' \in \{1, \ldots, R_\nu\}. \tag{4.14}$$

Let the corresponding index-sets of scenarios be $S_1, \ldots, S_{R_\nu}$ where $S_r := \{s \mid \omega^s \in \Omega_r,\ s \in S\}$. The corresponding *cell probabilities* are evaluated as $P_r := \sum_{s \in S_r} p_s$ where $\sum_{r=1}^{R_\nu} P_r = 1$. Under the cell-conditional first and second moments computed by:

$$\frac{1}{P_r} \sum_{s \in S_r} p_s \omega_k^s [\omega_l^s]^\theta, \quad \theta = 0, 1;\ k, l = 1, \ldots, K + L, \tag{4.15}$$

One obtains the upper and lower (second-order) scenario distribution approximations for cell Ω_r, using a procedure similar to (4.1)–(4.6), herein denoted by $\{(\hat{\omega}^i(r), \rho_i(r)) \mid i = 1, \ldots, I\}$ and $\{(\tilde{\omega}^i(r), \rho_i(r)) \mid i = 1, \ldots, I\}$, respectively. The monotonicity of the approximations under this partitioning procedure is given by the result below.

Theorem 4.2 *Under the above simplicial partition of Ω, the bounds behave monotonously for all $x \in X$, i.e.,*

$$\psi(x) \ \leq \ \psi_U^\nu(x) := \sum_{r=1}^{R_\nu} P_r \phi_U^{\nu,r}(x) \ \leq \ \sum_{i=1}^{K+L+1} \rho_i \phi(x, \hat{\omega}^i) \tag{4.16}$$

and

$$\psi(x) \ \geq \ \psi_L^\nu(x) := \sum_{r=1}^{R_\nu} P_r \phi_L^{\nu,r}(x) \ \geq \ \sum_{i=1}^{K+L+1} \rho_i \phi(x, \tilde{\omega}^i) \tag{4.17}$$

where

$$\phi_L^{\nu,r}(x) := \sum_{i=1}^{K+L+1} \rho_i(r)\phi(x,\tilde{\omega}^i(r)) \quad \textit{and} \quad \phi_U^{\nu,r}(x) := \sum_{i=1}^{K+L+1} \rho_i(r)\phi(x,\hat{\omega}^i(r)). \tag{4.18}$$

Corresponding to the above sequence of partitioning, one may obtain the sequence of upper and lower approximating objective value-sequences $\{Z_U^\nu\}$ and $\{Z_L^\nu\}$ by solving the linear programs

$$Z_U^\nu := \min_{x\in X}\{< c,x > + \psi_U^\nu(x)\} \quad \text{and} \quad Z_L^\nu := \min_{x\in X}\{< c,x > + \psi_L^\nu(x)\}, \tag{4.19}$$

along with their optimal first-stage solutions x_U^ν and x_L^ν, respectively. Due to Theorem 4.2, the sequence $\{Z_U^\nu\}$ is monotonically decreasing while $\{Z_L^\nu\}$ is monotonically increasing. Furthermore, as long as $\max_{r=1,\ldots,R_\nu} P_r \to 0$ when $\nu \to \infty$, the approximating scenario distributions $\rho(r)|_{r=1}^{R_\nu}$ can be shown to converge *weakly* to the original distribution $p_s|_{s=1}^{S}$. This yields $\lim_{\nu\to\infty} Z_U^\nu = Z^*$ and $\lim_{\nu\to\infty} Z_L^\nu = Z^*$, where Z^* is the optimal objective value of GLP_2 in (2.11).

Would the approximate x-solutions converge to true optimal solutions of GLP_2 in (2.11)? The answer to this question is provided by the theory of *epiconvergence*, which can loosely be described as the convergence of the epigraphical sets of the approximating functions (convex in x, in our case) to the epigraph of the original recourse function $\psi(.)$. With weak convergence of the approximating measures along with the joint domain Ω being a (compact) simplex, due to Birge and Wets [8, section 2.11], one has the required epiconvergence of the approximating functions. Under epiconvergence, Theorem 9 of Wets [72] shows that the set of approximate solutions belongs to the set of optimal solutions, i.e.,

$$\limsup[\operatorname{argmin} g^\nu] \subset \operatorname{argmin} g$$

where $g^\nu(x) := c'x + \psi_U^\nu(x)$ (respectively, $g^\nu(x) := c'x + \psi_L^\nu(x)$) and $g(x) := c'x + \psi(x)$. Hence, $\{x_U^\nu\} \to x^*$ and $\{x_L^\nu\} \to x^{**}$ hold as desired, where x^* and x^{**} are optimal solutions of the stochastic program (2.11).

There are at least two related issues: stability and speed of convergence. While it follows from the preceding results that the approximations may be sharpened to a user-specified accuracy through a partitioning procedure, that the solution sets so-generated behave in a continuous manner with respect to small perturbations in the approximating measures can be ensured by appealing to the stability results due to Kall [46] and Robinson and Wets [64]. The second important issue is how fast the convergence (or a given accuracy for the upper and lower approximations) can be achieved. Obviously, this depends on many factors such as the initial simplex used to contain the given set of scenarios, the manner in which simplicial decomposition is performed, as well as how efficiently one can solve the second-order approximations. These are the main topics under discussion of this chapter, as given in Section 4.2.

It is important to note that solving the upper and lower approximations in (4.19) for separate first-stage solution vectors x_U^ν and x_L^ν does not help much in identifying cells (i.e., clusters of scenarios) of the current partition for further refinement. Observing that the lower approximating solution x_L^ν is feasible in the upper approximating problem, the standard strategy is to compute a weaker upper bound by

$$\bar{Z}_U^\nu :=< c, x_L^\nu > \; + \psi_U^\nu(x_L^\nu). \tag{4.20}$$

Consequently, the accuracy of the second-order approximation for estimating the objective value Z^* of (2.11), at some iteration ν, can be evaluated on the basis of the current lower

bounding solution x_L^ν by (provided $Z_L^\nu > 0$)

$$\left| \frac{\bar{Z}_U^\nu - Z_L^\nu}{Z_L^\nu} \right| < \varepsilon, \tag{4.21}$$

i.e., if the *relative gap* of the upper and lower bounds for the current solution x_L^ν is within a certain tolerance ε, then no further refinement of the current approximation is needed.

4.2 Simplicial partitioning for iterative approximation

To apply the simplicial approximation technique presented in the preceding section, an initial ($K + L$ dimensional) simplex which can contain the given set of scenarios is needed. Usually, in practice, scenarios are constructed by following random variable outcome-trees, see for instance Cariño et al [10], and thus, when the scenarios are viewed as points in the $(K + L)$-dimensional space, their convex hull C may be far from being simplicial. Thus, an initial simplex Ω covering this convex hull is generally expected to consist of a larger volume of *zero-measure* which typically contributes to weakening the approximations. However, when refining the approximations through a simplicial decomposition of this initial simplex Ω, one is able to iteratively remove the areas of zero-measure, a procedure later described as *cell-redefining*.

4.2.1 Constructing an initial simplex Ω

Since the *empty spaces* in Ω adversely affect the tightness of the approximations, hence the speed of convergence, it is desirable to have an initial simplex as compact as possible. While many nonlinear programming formulations can be developed for this purpose, such an approach becomes an onerous task in itself when the scenario set is large in number.

Alternatively, one might determine a compact rectangle $\mathcal{R}$ containing $\mathcal{C}$ by defining:

$$\alpha_j := \min_{s \in S} \left\{ \omega_j^s \right\} \quad \text{and} \quad \beta_j := \max_{s \in S} \left\{ \omega_j^s \right\}, \quad \forall j = 1, \ldots, K+L, \tag{4.22}$$

and setting $\mathcal{R} := \times_{j=1}^{K+L} [\alpha_j, \beta_j]$. Subsequently, a simplicial decomposition of this rectangle may be attempted since $\mathcal{C} \subseteq \mathcal{R}$. However, this leads to increasing the number of simplices in the problem exponentially, and thus, may not be desirable from a computational point of view. Instead, we take a simplistic approach by adopting from Horst and Tuy [41] and modifying as follows: Let

$$\beta_0(\delta) := \max_{s \in S} \left\{ \sum_{j=1}^{K+L} \delta_j \omega_j^s \right\} \tag{4.23}$$

for some fixed coefficient vector $\delta \in \Re^{K+L}$ such that $\delta_j > 0$, $j = 1, \ldots, K+L$. Then, $\Omega(\delta)$ defined by

$$\Omega(\delta) := \left\{ \omega \in \Re^{K+L} \; : \; \sum_{j=1}^{K+L} \delta_j \omega_j - \beta_0(\delta) \leq 0, \; \omega_j - \alpha_j \geq 0, \forall j = 1, \ldots K+L \right\} \tag{4.24}$$

is a simplex containing $\mathcal{C}$. Furthermore, this simplex is characterized by the set of extreme points $\mathcal{V} := \{w^1, w^2, \ldots, w^{K+L+1}\}$ where the j^{th} component of w^1 is $w_j^1 = \alpha_j$ for $j = 1, \ldots, K+L$ and the remaining vertices w^i (for $i = 2, \ldots, K+L+1$) are given by:

$$w_j^i(\delta) = \begin{cases} \alpha_j & \text{if } i \neq j+1 \\ \dfrac{1}{\delta_j} \left[\beta_0(\delta) - \displaystyle\sum_{k=1, k \neq j}^{K+L} \delta_k \alpha_k \right] & \text{if } i = j+1. \end{cases} \tag{4.25}$$

As a result, the *degree of uncertainty* – defined as the maximum variability of the values that the random variables can take for a given domain – has increased when considering $\Omega(\delta)$ instead of $\mathcal{C}$. While this leads to the effect of diluting the quality of approximations, the cell-redefining strategy discussed in the next section may be used to reduce the degree of uncertainty under a simplicial decomposition of $\Omega(\delta)$.

Nevertheless, the coefficient vector δ may be chosen to control the effect of the initial simplex on the stochastic program. Typically, the uncertainty in the objective coefficients has a lesser effect on the tightness of the approximations (as will be demonstrated in the computational section) than that in the right-hand side coefficients of the recourse linear program. This motivates one to construct the simplex $\Omega(\delta)$ such that the increase in the degree of uncertainty in the ξ components is made smaller than that in the η components. This may be accomplished using the following simple result:

Proposition 4.3 *For e_j denoting the j^{th} elementary vector and $\delta \in \Re^{K+L}$ being a given fixed coefficient vector such that $\delta > 0$, $w_j^{j+1}(\delta + \epsilon e_j) \leq w_j^{j+1}(\delta)$ holds for every $\epsilon > 0$.*

Proof. Rewriting, $\beta_0(\delta) = \max_{s \in S}\{C_s + \delta_j \omega_j^s\}$ where we have defined $C_s := \sum_{k \neq j, k=1}^{K+L} \delta_k \omega_k^s$. Also define $G := \sum_{k \neq j, k=1}^{K+L} \delta_k \alpha_k$. Clearly, $G \leq C_s$ for all $s \in S$. Hence,

$$\begin{aligned} w_j^{j+1}(\delta) &= \max_{s \in S}\{C_s/\delta_j + \omega_j^s\} - G/\delta_j \\ &= \max_{s \in S}\{(C_s - G)/\delta_j + \omega_j^s\} \\ &\geq \max_{s \in S}\{(C_s - G)/(\delta_j + \epsilon) + \omega_j^s\} = w_j^{j+1}(\delta + \epsilon e_j) \end{aligned}$$

holds since $(C_s - G) \geq 0$, $\forall s \in S$. ■

Accordingly, one would be motivated to assign smaller δ_j for coordinates along which the underlying function is closer to being linear. However, since the recourse function ϕ depends on first stage decisions x as well, implementation of this idea is somewhat difficult. In Section 4.4, we discuss this sensitivity of δ on the tightness of the approximations.

4.2.2 Cell-Redefining (C-R) procedure

A C-R procedure was introduced and implemented in [25] in order to further tighten first-order bounds when partitioning in rectangular domains. In this section, we pursue a similar

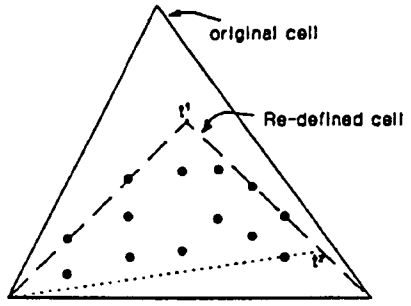


Figure 4.2: Cell-redefining procedure for a simplex.

idea with regard to simplicial domains.

Let us consider a simplicial partition of the initial simplex Ω, as indicated in (4.14). In particular, at some iteration ν of partitioning, for some $r \in \{1, \ldots, R_\nu\}$, the cell Ω_r is a simplex itself in $\Re^{K+L}$. The set of scenarios in this cell is described by $\{\omega^s : s \in S_r\}$. Let $C_r := conv\{\omega^s : s \in S_r\}$, the convex hull.

Definition 4.4 *The cell Ω_r is said to have removable volume of zero measure if there exists a simplex Ω'_r such that $C_r \subseteq \Omega'_r \subset \Omega_r$.*

The C-R procedure strives for such a simplex Ω'_r (with the least possible volume) for a given cell Ω_r. The motivation is that the upper and lower approximations on Ω'_r will be much tighter than those on Ω_r. See Figure 4.2 for an illustration in $\Re^2$.

For the convenience of exposition, let us drop the index r. The set of extreme points of Ω is $\{w^1, \ldots, w^{K+L+1}\}$ and the associated vertex matrix is denoted by V, see (4.1). The cell-redefining procedure selects a vertex w^j of Ω, checks if this vertex can be replaced (i.e., redefined) by a point $t \in \Omega$ such that the simplex

$$\Omega' := conv\{w^1, \ldots, w^{j-1}, t, w^{j+1}, \ldots, w^{K+L+1}\} \subset \Omega, \tag{4.26}$$

and, if so, replaces Ω by Ω' provided that $\Omega \setminus \Omega'$ is of sufficiently large volume.

For each scenario ω^s $(s \in S)$ in Ω, determine the *convex multipliers* $\lambda^s (\in \Re^{K+L+1})$ by

$$\lambda^s := V^{-1} \begin{pmatrix} \omega^s \\ 1 \end{pmatrix} \tag{4.27}$$

and thus $0 \leq \lambda_i^s \leq 1$ and $\sum_{i=1}^{K+L+1} \lambda_i^s = 1, \forall s \in S$. The set of multipliers associated with vertex w^i is represented by $\Lambda_i := \{\lambda_i^s \ : \ s \in S\}$. Observe that $\mu_i := \max\{\Lambda_i\}$ is a *measure of closeness* of the scenarios (in the cell) to a given vertex w^i. If μ_i is close to one, then of course, one would not expect a significant reduction in the volume of Ω when the vertex w^i is redefined. In fact, the *best* vertex for redefining, in this sense, is determined by the index

$$j := \arg\min\{\mu_i \ : \ i = 1, \ldots, K+L+1\}. \tag{4.28}$$

Therefore, unless $\mu_j < \mu^*$, a user-specified threshold value (< 1), the C-R procedure should not be invoked.

Suppose $\mu_j < \mu^*$, and thus, the vertex w^j is to be replaced by the $(K+L)$-dimensional point $t \in \Omega$ such that t is *furthest* as possible from w^j, in the sense of the associated multipliers, such that (4.26) holds. This problem may be formulated as:

$$\begin{aligned} \min_{t \in \Re^{K+L}} \quad & \left[V^{-1} \begin{pmatrix} t \\ 1 \end{pmatrix} \right]_j \\ \text{s.t.} \quad & V^{-1} \begin{pmatrix} t \\ 1 \end{pmatrix} \geq 0 \\ & \hat{V}^{-1} \begin{pmatrix} \omega^s \\ 1 \end{pmatrix} \geq 0, \ \forall s \in S \end{aligned} \tag{4.29}$$

where $\hat{V}$ is the resulting matrix when the j^{th} column of V is replaced by the column-vector

$\begin{pmatrix} t \\ 1 \end{pmatrix}$. The inverse of $\hat{V}$ is obtained by $\hat{V}^{-1} = UV^{-1}$ where U is the *eta-matrix* given by

$$U := \begin{bmatrix} 1 & 0 & \cdots & 0 & -\frac{\hat{t}_1}{\hat{t}_j} & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 & -\frac{\hat{t}_2}{\hat{t}_j} & 0 & \cdots & 0 \\ \vdots & & & & \vdots & & & \vdots \\ 0 & 0 & \cdots & 0 & \frac{1}{\hat{t}_j} & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & -\frac{\hat{t}_{j+1}}{\hat{t}_j} & 1 & \cdots & 0 \\ \vdots & & & & \vdots & & & \vdots \\ 0 & 0 & \cdots & 0 & -\frac{\hat{t}_{K+L+1}}{\hat{t}_j} & 0 & \cdots & 1 \end{bmatrix} \tag{4.30}$$

and

$$V^{-1} \begin{pmatrix} t \\ 1 \end{pmatrix} =: \hat{t} \in \Re^{K+L+1}. \tag{4.31}$$

Upon algebraic manipulation and simplification, (4.29) reduces to the following linear program:

$$\begin{aligned} \min_{t \in \Re^{K+L}} \quad & V^{-1(j)} \begin{pmatrix} t \\ 1 \end{pmatrix} \\ \text{s.t.} \quad & V^{-1} \begin{pmatrix} t \\ 1 \end{pmatrix} \geq 0 \\ & \left(\lambda_i^s V^{-1(j)} - \lambda_j^s V^{-1(i)}\right) \begin{pmatrix} t \\ 1 \end{pmatrix} \geq 0, \forall i \neq j,\ i = 1, \ldots K+L+1;\ s \in S, \end{aligned} \tag{4.32}$$

where $V^{-1(j)}$ refers to the j^{th} row of V^{-1}. The LP in (4.32) has $(K+L)$ variables and $(|S|+1)(K+L)+1$ constraints. Thus, the dual problem, which is referred to as CRLP and has only $(K+L)$ constraints, is easier to solve generally provided $|S|$ is not too large. Initially, when the number of scenarios under consideration is large, therefore, cell-redefining should not be invoked; besides, in the initial iterations, given the way the initial simplex is constructed, cell-redefinition may not yield a significant reduction of the volume of zero measure. By setting μ^* small enough, one may avoid the latter situation. However, as the simplicial partition progresses, resulting cells (Ω_r) will begin to contain lesser number of

scenarios, i.e., $|S_r|$ will be smaller, thus offering a better chance of being redefined to tighten the approximations.

4.2.3 Partitioning strategies

At the current iteration ν, a cell for further partitioning is chosen by the maximum (probabilistically) weighted difference (WDIFF) criterion, i.e.,

$$\max_{r=1,\ldots,R_\nu} P_r \left[\phi_U^{\nu,r}(x_L^\nu) - \phi_L^{\nu,r}(x_L^\nu)\right]. \tag{4.33}$$

When only a single cell is further partitioned at some iteration, it is referred to as *single partitioning*. However, when partitioning multiple cells, referred to as *multiple partitioning*, we pick all cells having WDIFF within 60% (rule of thumb) of the maximum in (4.33). This is typically done in order to save computations when the lower bounding solutions x_L^ν are stabilized, as indicated by the percent change in x solutions (for 2-norm):

$$\left|\frac{||x_L^{\nu+1}|| - ||x_L^\nu||}{||x_L^\nu||}\right| \times 100\%. \tag{4.34}$$

Given a cell chosen for further sub-division, one may proceed in at least two ways: *radial partitioning* or *bi-partitioning*. In radial partitioning of a cell Ω_r, a point interior to Ω_r is chosen and it is joined to all vertices of Ω_r to generate $(K + L + 1)$ sub-simplices. While increasing the number of cells in this manner leads to rapid growth of the size of the lower approximation, if implemented in the initial partitioning iterations, it may also adversely affect the convergence behavior of the x-solutions due to the way in which partitioning is performed. Furthermore, it can over-partition a cell and lead to wasteful computation effort. Therefore, we resort to the bi-partitioning procedure in which a cell Ω_r is split into only two sub-simplices by choosing a single *partition plane* that passes through vertices, dividing an edge. In so doing, thus, one needs to address:

1. the selection of the partitioned edge, and

2. the point at which the selected edge is to be partitioned.

Bisection partitioning

The simplest strategy in partitioning is to choose the edge with the longest length and then partition it at the mid-point. That is, if the longest edge is determined by the two vertices (w^i, w^j), it will be partitioned at $\hat{w} := (w^i + w^j)/2$ under bisection partitioning, herein referred to as the Δ_0-strategy. This leads to two new simplicial cells after partitioning given by:

$$\begin{aligned} &conv\{w^1, \ldots w^{i-1}, \hat{w}, w^{i+1}, \ldots w^{K+L+1}\} \text{ and} \\ &conv\{w^1, \ldots w^{j-1}, \hat{w}, w^{j+1}, \ldots w^{K+L+1}\}. \end{aligned} \tag{4.35}$$

Note that the corresponding inverses of the two vertex-matrices can be obtained by single-pivot updates of the inverse V^{-1} of the cell just partitioned. The general motivation for bisection partitioning is that cells can be driven uniformly to very small sizes, thereby assuring theoretical convergence.

Partitioning based on nonlinearity

Instead of relying on (a passive strategy such as) arbitrarily small cell-sizes, by using more efficient partitioning directions based on the behavior of the recourse function in the chosen cell, one may hope to accelerate the convergence process. Recalling that if the function is linear on a cell, then the approximations provide exact expected value, the motivation is to evaluate the *degree of nonlinearity* of the recourse function along edges of Ω_r, see [25] and [36], for instance. However, since both ξ and η coordinates may vary along certain

edges, the standard procedure in the latter references for evaluating nonlinearity cannot be employed.

Let us drop the index r of the chosen cell Ω_r in the current partitioning iteration ν. Also drop the iteration index ν from the current lower bounding solution x_L^ν that is being used. Suppose the vertices $w^i \equiv (u^i, v^i)$ and $w^j \equiv (u^j, v^j)$ represent the edge to be divided and that $u^i \neq u^j$ and $v^i \neq v^j$. We employ one of two possible methods in such cases to measure the degree of nonlinearity associated with the edge (i, j).

The first measure is based on the directional derivative information. Let D_{ij} denote the directional derivative of the recourse function at w^i along the direction $w^j - w^i$. Similarly, D_{ji} is also defined. To determine D_{ij}, let (y^i, π^i) be the optimal primal and dual solutions of the recourse linear program determining $\phi(x_L, w^i)$. Then

$$D_{ij} = (y^i, \pi^i)\frac{d_{ij}}{|d_{ij}|} \tag{4.36}$$

where

$$d_{ij} := \begin{pmatrix} q(v^j) - q(v^i) \\ h(u^j) - T(u^j)x_L - h(u^i) + T(u^i)x_L \end{pmatrix}. \tag{4.37}$$

Definition 4.5 *The local nonlinearity measure Δ_1 for the edge (i, j) is defined by*

$$\Delta_1(i, j) := |D_{ij} + D_{ji}|. \tag{4.38}$$

The motivation for the measure Δ_1 is that if the recourse function is linear on the edge (i, j) then $\Delta_1(i, j) = 0$. Consequently, $\Delta_1(i, j) \neq 0$ implies that the recourse function is nonlinear along the edge (i, j).

The second measure of nonlinearity, denoted $\Delta_2(i, j)$, considers the displacement from w^i to w^j in two possible two-step movements: $w^i \to \hat{w} := (u^j, v^i) \to w^j$ and $w^i \to \tilde{w} :=$

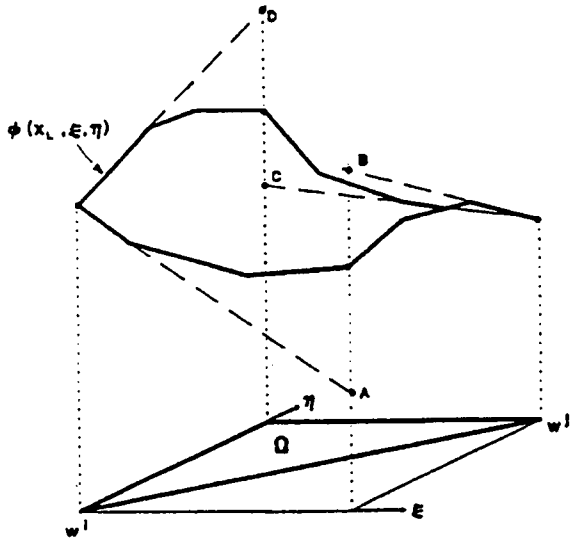


Figure 4.3: Nonlinearity measure $\Delta_2(i,j) = \min[\text{AB, CD}]$.

$(u^i, v^j) \rightarrow w^j$. Then, for each possible two-step displacement, the recourse function is approximated by two-piece linear functions, determined by primal and dual solutions of the recourse problem at w^i and w^j. Then the functional displacement due to this two-piece linear approximation is computed at both $\hat{w}$ and $\tilde{w}$, of which the minimum is taken as the nonlinearity measure; see Figure 4.3 for an illustration in $\Re^2$. That is, denoting the optimal primal and dual solutions corresponding to $\phi(x_L, w^i)$ and $\phi(x_L, w^j)$ by (y^i, π^i) and (y^j, π^j), respectively, we have

$$\phi(x_L, \hat{w}) \geq \pi^i[h(u^j) - T(u^j)x_L] \quad \text{and} \quad \phi(x_L, \hat{w}) \leq q(v^i)'y^j. \tag{4.39}$$

Furthermore,

$$\phi(x_L, \tilde{w}) \geq \pi^j[h(u^i) - T(u^i)x_L] \quad \text{and} \quad \phi(x_L, \tilde{w}) \leq q(v^j)'y^i. \tag{4.40}$$

Consequently, the second measure of nonlinearity for the edge (i,j) is defined by

$$\Delta_2(i,j) := \min\left\{q(v^i)'y^j - \pi^i[h(u^j) - T(u^j)x_L]\,,\; q(v^j)'y^i - \pi^j[h(u^i) - T(u^i)x_L]\right\}. \tag{4.41}$$

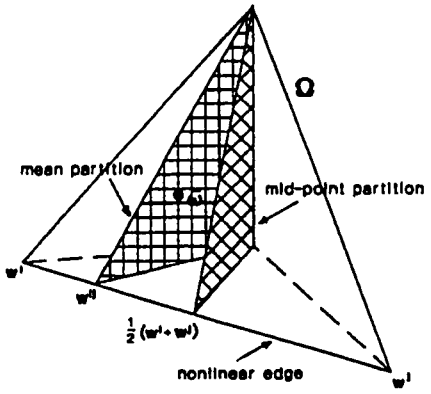


Figure 4.4: Conditional-mean and mid-point partitioning.

The intuition behind the measure Δ_2 is that if the recourse function is almost linear in the rectangle determined by the points $w^i, \hat{w}, w^j$ and $\tilde{w}$, then Δ_2 is expected to be smaller. More importantly, if either $u^i = u^j$ or $v^i = v^j$ occurs, then the nonlinearity measure Δ_2 reduces to that proposed in [36].

Conditional-mean partitioning

Once an edge (i, j) is selected – corresponding to the maximum of the chosen nonlinearity measure – one can proceed with partitioning the current simplex either by bisecting the edge (i, j) or determining a partitioning plane based on probability mass. The former approach is referred to as mid-point partitioning and is identified by $\hat{\Delta}_1$ and $\hat{\Delta}_2$.

In the second approach, in order to balance the probability mass in the resulting two cells after partitioning, one is motivated to determine a partitioning plane that passes through the conditional mean (denoted by $\bar{\omega}$) for the current cell Ω. (See Figure 4.4.) This practice is consistent conceptually with that in rectangular domains. The following result is useful in accomplishing this objective.

Proposition 4.6 *For the $(K+L)$-dimensional simplex Ω with vertices w^k, $k = 1, \ldots, K+L+1$, let ρ be defined by (4.2) for the first moments $\bar{\omega}$. Then, the hyperplane in $\Re^{K+L}$ determined by $\bar{\omega}$ and the $(K+L-1)$ vertices w^k $(k = 1, \ldots, K+L+1,\ k \neq i, k \neq j)$ intersects the edge (i,j) at*

$$w^{ij} := \frac{\rho_i}{\rho_i + \rho_j} w^i + \frac{\rho_j}{\rho_i + \rho_j} w^j. \tag{4.42}$$

Proof. Let the partitioning plane in the proposition be represented by $a'\omega = b$ for coefficients $a \in \Re^{K+L}$ and $b \in \Re$. Thus, we have

$$a'\bar{\omega} = b, \quad \text{and} \quad a'w^k = b,\ \forall k = 1, \ldots, K+L+1,\ k \neq i, k \neq j. \tag{4.43}$$

Multiplying the last $(K+L-1)$ equalities by the corresponding ρ_k and summing up,

$$\sum_{k \neq i,j; k=1}^{K+L+1} \rho_k [a'w^k] = \sum_{k \neq i,j; k=1}^{K+L+1} \rho_k b. \tag{4.44}$$

Subtracting (4.44) from the first equality in (4.43) and rearranging, $a'w^{ij} = b$ follows where w^{ij} is defined in (4.42). Thus, w^{ij} belongs to the hyperplane of interest. Moreover, w^{ij} belongs to the edge (i,j), which completes the proof. ■

The strategy that utilizes such conditional-mean partitioning planes is referred to as $\bar{\Delta}_1$ and $\bar{\Delta}_2$, corresponding to the nonlinearity measure.

4.3 Solution procedure and implementation

The implementation of the Second Moment Approximation for lineaR sTochastic programs (with two decision-stages) - SMART- routine involves designing several modular routines to perform the tasks needed in the solution procedure. These modules include the SOSA (Second-Order Scenario Approximation) module, the BOUNDS module for computing bounds, and the SIMDEC module comprising Procedure SP (Simplicial Partitioning)

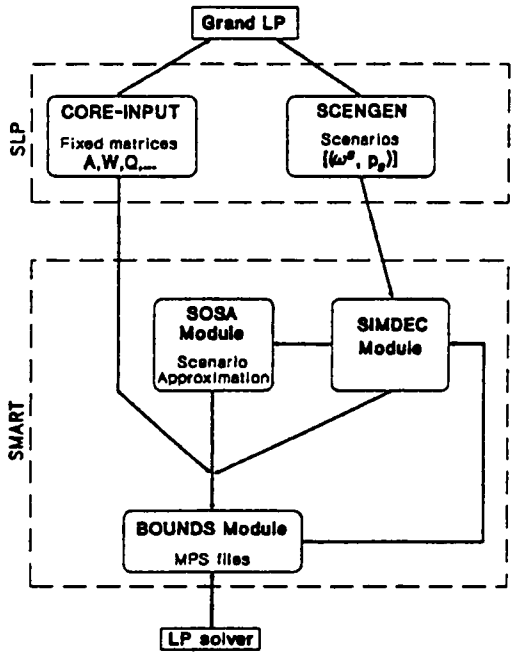


Figure 4.5: Schematic diagram of the SMART modules.

and Procedure CR (Cell-Redefining.) As a supplementary module for generating SLP's, the SLP module consists of Kall-Keller [48] GENSLP code for generating core input data and SCENGEN module for generating stochastic data. The general scheme of modular interaction is depicted in Figure 4.5. In what follows, a brief description of SMART modules is provided.

Given a stochastic linear program (SLP) with its *core input data*: c, b, A, W, $T_0, \ldots T_K$, $h_0, \ldots h_K$, and $q_0, \ldots q_L$, and *stochastic data*: the scenario set $\{(\omega^s, p_s) : s \in S\}$, an initial simplex Ω enclosing the set of scenarios is constructed according the procedure described in section 4.2.1. In particular, it allows sensitivity analysis with respect to the initial simplex parameter δ_0, by fixing δ in (4.24) as follows:

$$\delta_j = \begin{cases} \delta_0 & \text{if } j = 1, \ldots, K \\ 1 & \text{if } j = K+1, \ldots, K+L. \end{cases} \tag{4.45}$$

The general solution procedure of SMART is described in the following.

Step 1 Perform second-order scenario approximation (SOSA).

Given the geometric information of newly created cells in the simplicial partition, as well as the clustering of scenarios from SIMDEC, the SOSA module first computes the required second-moment information for each new cell Ω_r. Consequently, according to (4.1)-(4.6), the lower and upper approximating scenario distributions are computed as outputs.

Step 2 Compute bounds (BOUNDS).

For the fixed data from CORE-INPUT, using the lower approximation from SOSA and the geometric information from SIMDEC, the BOUNDS module first generates the lower bounding linear program LBLP in MPS format. Then an LP solver is called – in our case, IBM's OSL Subroutine Version 1.2 – to solve this LP and the lower bounding solution x_L^ν is obtained.

Then, for this first-stage solution x_L^ν, combining with CORE-INPUT data and information from SOSA and SIMDEC, the upper bounding LPs for each cell Ω_r are formed in MPS format and solved successively. The objective values $\phi_U^{\nu,r}(x_L^\nu)$ thus obtained, see (4.18), are accumulated for each cell and weighted by the associated probabilities P_r, $r = 1, \ldots, R_\nu$, to obtain the upper bound $\bar{Z}_U^\nu$ in (4.20).

Subsequently, the relative gap test is performed according to (4.21) to determine if termination should occur for the specified ε. If not, the current solution x_L^ν, as well as the weighted difference WDIFF of upper/lower bounds for all the current cells become the input to the SIMDEC module, and the iteration is continued.

Step 3 Refine the current partition (SIMDEC).

This module performs the simplicial decomposition of the domain Ω, assigns scenarios to cells of the decomposition, and maintains and updates the geometric information of cells from one partitioning iteration to another. When invoked with a current solution iterate x_L^ν, this module determines

1. a cell – denoted herein by Ω_r at some partitioning iteration ν – for further partitioning, according to the rule (4.33),
2. an edge (w^i, w^j) identified by a partitioning strategy such as Δ_0, Δ_1, or Δ_2, and
3. a partitioning point w^* of the latter edge, i.e., either the mid-point $(w^i + w^j)/2$ or the point w^{ij} corresponding to the conditional-mean partitioning plane, see (4.42).

A chosen cell Ω_r is partitioned into two new cells Ω_r^1, and Ω_r^2 and the information is updated according to the following procedure:

Procedure-SP:
begin
$\mathcal{S}^\nu := \{\mathcal{S}^{\nu-1} \setminus \Omega_r\} \cup \{\Omega_r^1, \Omega_r^2\}$
$\mathcal{V}_r^1 := \{\mathcal{V}_r \setminus \{w^i\}\} \cup \{w^*\}$;
$\mathcal{V}_r^2 := \{\mathcal{V}_r \setminus \{w^j\}\} \cup \{w^*\}$;
$(V_r^1)^{-1} = U_i V_r^{-1}$;
$(V_r^2)^{-1} = U_j V_r^{-1}$;
$S_r^1 = \emptyset$;
$S_r^2 = \emptyset$;
for all $s \in S_r$ **do**
$\lambda\text{temp} = U_i \lambda^s$;
if $\lambda\text{temp} \geq 0$ **then begin**
$\lambda^s = \lambda\text{temp}$;
$S_r^1 = S_r^1 \cup \{s\}$;
else begin
$\lambda^s = U_j \lambda^s$;
$S_r^2 = S_r^2 \cup \{s\}$;

end;(*if*)
end;(*for*)
end;

For the cell Ω_r, U_i is the *eta matrix* with its i^{th} column being the *eta column*

$$(\frac{-t_1}{t_i}, \ldots, \frac{-t_{i-1}}{t_i}, \frac{1}{t_i}, \frac{-t_{i+1}}{t_i}, \ldots, \frac{-t_{K+L+1}}{t_i})' \quad \text{where} \quad t = V_r^{-1}(w^{*\prime}, 1)'. \qquad (4.46)$$

The Procedure-SP can be invoked more than once before leaving the SIMDEC module, as determined by multiple cell partitioning, see (4.33) and the discussion that follows.

Cell-Redefining (C-R):

The C-R procedure is a part of the SIMDEC module for redefining the vertices of cells obtained through the simplicial decomposition. As described in Section 3.2, each new cell Ω_r created in the current partitioning step is *admissible* for redefining provided that the corresponding CRLP, i.e., the dual of (4.32), does not have too many variables, i.e., $|S_r|(K+L) \leq 1000$. For an admissible cell, the vertices that will be attempted for redefining are selected by the criterion $\mu_j \leq \mu^*$ (where the threshold μ^* is set to be 0.70, see Section 4.22), by not doing so, the chances of obtaining a significant reduction in the cell volume are slim.

The Procedure-CR below, also (heuristically) tests the *replaceability* of the chosen vertex w^j by checking if there are scenarios on the facets (incident to w^j) describing the simplex, if so, one would not expect to obtain a significant reduction in the size of the current simplex.

The input to Procedure-CR is the set of multipliers $\Lambda_i := \{\lambda_i^s : s \in S_r\}$, for $i = 1, \ldots, K+L+1$ and $\mu_i := \max \Lambda_i$. λ's are already available within SIMDEC. Denote the potential set of vertices for redefining by $J_0 := \{i : \mu_i \leq \mu^*, \mu_i > 0\}$. For a

chosen vertex w^j, denote the optimal value of (4.32) by $CRLP(w^j) := V^{-1(j)} \begin{pmatrix} t^* \\ 1 \end{pmatrix}$, where t^* is the point that replaces w^j.

Define J_1 as the index-set of the already re-defined vertices. Initialize $J_1 = \emptyset$.

Procedure-CR:

Step C-R 1: If $J_0 = \emptyset$, then STOP.

Step C-R 2: Find $j := \arg\min\{\mu_i : i \in J_0\}$.

Step C-R 3: Check *replaceability* of w^j as follows.

begin
  for $i = 1$ **to** $K + L + 1$ **do if** $i \neq j$ **then begin**
    if $\min \Lambda_i > 0$ **then** go to step C-R 4;
  end (*for*);
  go to step C-R 5;
end;

Step C-R 4: Solve dual of (4.32) to obtain $CRLP(w^j)$ and t^*.
If $CRLP(w^j) > 0$ then update the geometric information as follows.
begin
  $w^j = t^*$;
  $V_r^{-1} = U^j V_r^{-1}$;
  for all $s \in S_r$ **do**
    $\lambda^s = U^j \lambda^s$;
  end;
  Update $\Lambda_i := \{\lambda_i^s : s \in S_r\}$; and $\mu_i := \max \Lambda_i, \forall i = 1, \ldots K + L + 1$;
  $J_0 = \{i : \mu_i \leq \mu^*, \mu_i > 0, i = 1, \ldots, K + L + 1\}$
end; (U^j is the resulting eta-matrix when using (4.46) with t^* in place of w^*).

Step C-R 5: Update: $J_1 = J_1 \cup \{j\}$
$J_0 = J_0 \setminus \{J_0 \cap J_1\}$ and go to Step C-R 1.

4.4 Summary of computational results

The implementation in the preceding section was coded in FORTRAN 77 and developed on a SUN SparcStation 2 running under UNIX. IBM's OSL Version 1.2 was used as the subroutine for solving LPs.

The SMART code allows up to 16 (dependent) random variables (max: $K = 8, L = 8$) and up to 100,000 scenarios. The maximum number of simplicial partitions allowed is 20.

The core input has the limits: $m_1, n_1, m_2, n_2 \leq 100$ and the stochastic linear programs are generated by specifying a random seed for GENSLP and a second random seed for SCENGEN. The above limitations are set only due to memory restrictions of the machine.

All coefficients in the CORE-INPUT are generated uniformly from the interval $[-10, 10]$. For the purposes of the experiments reported here, all scenarios are generated such that the corresponding random variables have a maximum spread of $[-5, 5]$. Furthermore, only the complete-recourse matrix option is used in GENSLP along with matrices being specified to have 10% density.

In each category of experimentation, at least 5 problems have been solved and averaged when comparisons are made. Specifically, the following issues are investigated:

1. the sensitivity of the second-order simplicial approximations to the initial simplex (specified by the value of δ_0.)

2. the effect of joint partitioning compared to coordinate axis partitioning with respect to first-order approximations

3. the effect of using second-order information over first-order information when using joint-partitioning

4. the effect of partitioning strategy and multiple partitioning on second moment approximations.

The set of problem characteristics given in Table 4.1 serves as the base for the above investigations.

A summary of the computational results follows.

Table 4.1: Randomly generated 2-stage sample problem categories.

Problem category	K	L	(m_1, n_1, m_2, n_2)	# Scenarios
1	5	5	(20,40,20,40)	2048
2	5	2	(20,40,20,40)	2048
3	8	0	(10,20,10,20)	1024
4	1	1	(5,10,5,10)	[100,700][1]
5	5	5	(10,20,10,20)	1024
6	8	8	(30,50,30,50)	65536

(1) Regarding the sensitivity of the second-order simplicial approximations to the initial simplex, as shown in Figure 4.6 and Figure 4.7, increasing δ_0(reducing the degree of uncertainty in ξ) has the tendency to strengthen the second moment approximation in the initial iterations, but eventually this effect is diminished.

(2) Regarding the effect of using second-moment over first moments, test results for convex cases show that approximations using second-moment produce much faster convergence of objective values when compared to that using only first moments. Much of the adjustment in x_L occur in the initial iteration of SMART, while x solution continues to change until much later in first-order approximations. The result suggests that the convergence of x is faster in second-moment approximations.

(3) The effect of partitioning points using the conditioning mean partitioning point versus using the mid-point partitioning plane is depicted in Figure 4.8. Our computational experience suggests that mid-point partitions may be dominated by conditional-mean point partitions.

(4) Regarding the effect of multiple partitioning versus single partitioning, results show that multiple partitioning is a better procedure than single partitioning for saving

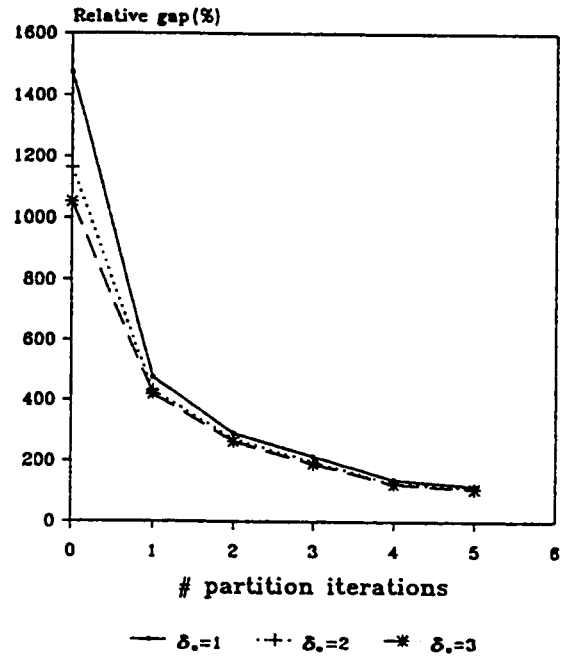


Figure 4.6: Sensitivity on the initial simplex (problem category #1.)

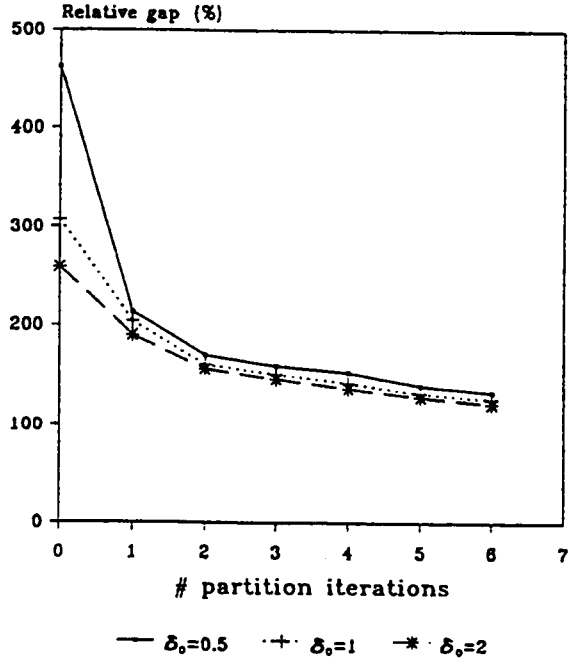


Figure 4.7: Sensitivity on the initial simplex (problem category #2.)

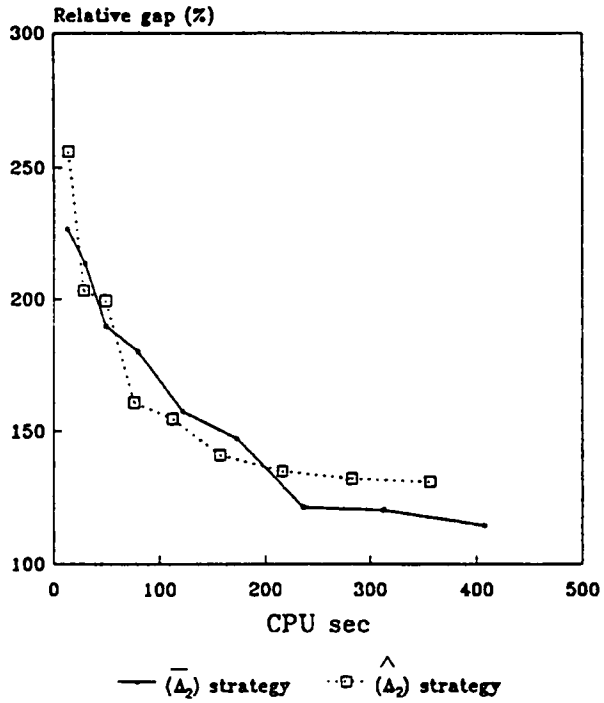


Figure 4.8: Effect of partitioning point (problem category #3.)

computational effort, especially useful after initial few iterations of SMART because x_L tends to be stablized after few iterations.

(5) The effect of cell-redefining strategy is illustrated in Figure 4.9. There is very little improvement brought about by the C-R strategy. But, after a significant refinement of the simplicial partition, the C-R strategy is able to effectively remove the areas of zero measure. Consequently, the approximations become tighter and convergence is achieved faster.

(6) The sensitivity of SMART solution time to the number of scenarios in the problem is plotted in Figure 4.10, and is compared with the CPU times for solving the grand LP. The results show that the solution time for grand LP is adversely affected by the number of scenarios, while the solution time for bounding LP's increases with the

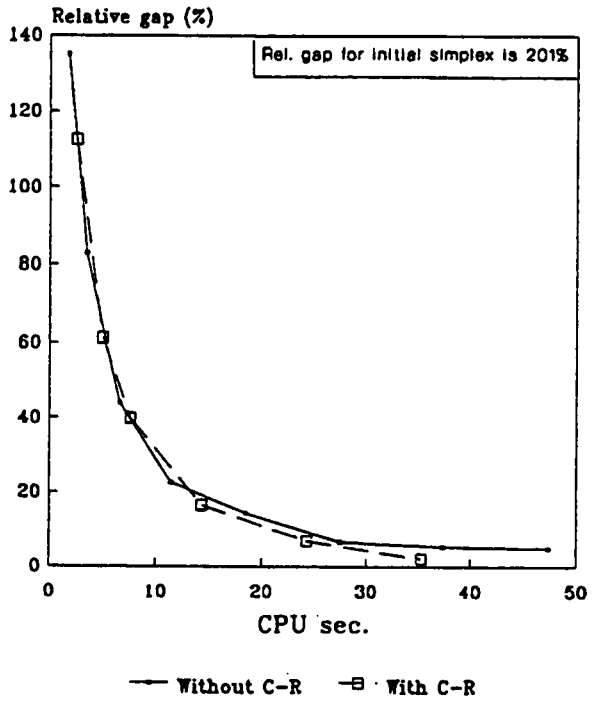


Figure 4.9: Effect of cell-redefining (problem category #4.)

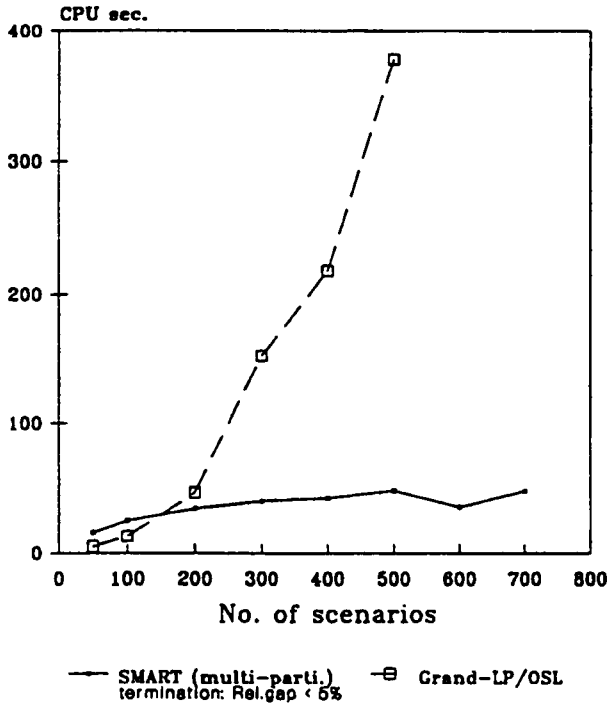


Figure 4.10: Sensitivity on number of scenarios (problem category #4.)

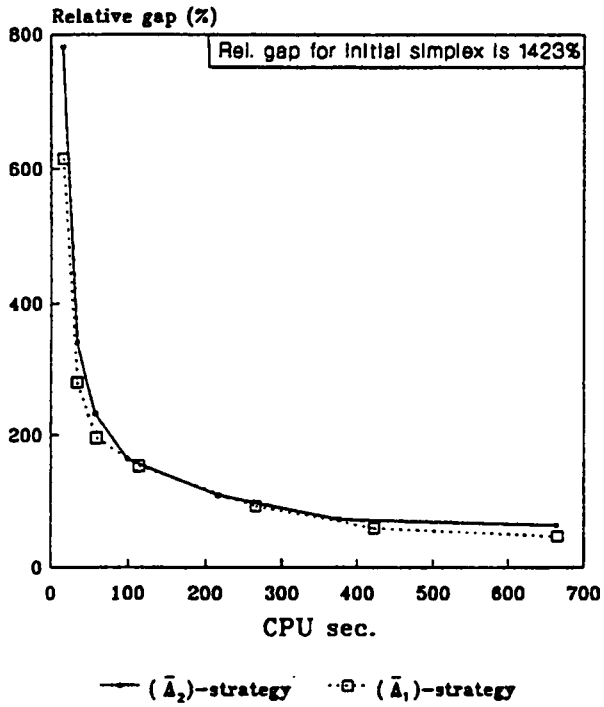


Figure 4.11: Effect of partitioning strategy (problem category #5.)

sizes of K and L. Nevertheless, increasing the dimensions K and L, while holding the number of scenarios constant, leads to larger volume of zero measure in the cells of simplicial partition. By using C-R strategy, one has better chance of removing areas of zero measures.

(7) The effect of partitioning strategies Δ_1 and Δ_2 combined with conditional-mean partitioning for convex-concave case (K and L are both non-zero) is presented in Figure 4.11. Our limited computational experience does not favor one over the other. When K and L are roughly about the same size, $\bar{\Delta}_1$ does seem to have a slight advantage over $\bar{\Delta}_2$.

(8) Computational comparisons of simplicial second-order moment approximations, simplicial first-order moment approximations, and rectangular first-order moment approxi-

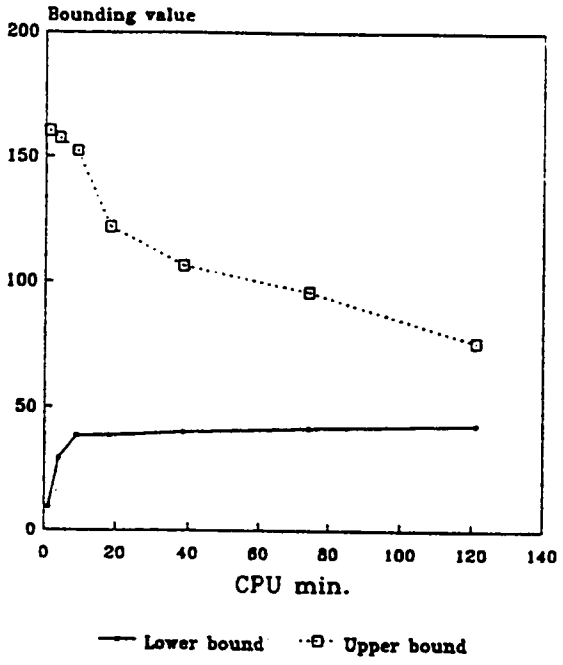


Figure 4.12: Behavior of SMART bounds.

mations, show simplicial second-order moment approximations result in better bounds which converge faster in terms of CPU time. The superiority becomes more evident when K and L are significantly large.

(9) The performance of SMART for a case of larger K and L is graphed in Figure 4.12. The general observation is that the second-order lower bounds tend to converge much faster than the upper bounds.

More details about the computational results of SMART can be found in Edirisinghe and You [21].

The computational results demonstrate the effectiveness of second-order simplicial approximations for solving two-stage stochastic linear programs. The use of second-order moments and effective partitioning strategies on simplicial joint domain yield the desired

computational efficiency in solving large two-stage stochastic programs. The second-order simplicial approximation solution procedure is thus proven to be a powerful alternative for solving two-stage stochastic linear programs. This provides sufficient motivation for further testing such solution procedure on standard problems and real-life problems. This also give motivation to extend the solution approach for solving multistage models.

Chapter 5

New Solution Methodology for Multi-stage Models

The solution methodology described in this chapter extends the two-stage scenario approximation method discussed in Chapter 4 to a multi-stage setting. Consider the multistage stochastic program GLP_T in (2.16) in which the uncertainty is represented by a set of future scenarios.

Recall that the two-stage scenario approximation method utilizes the second-order moment information and simplicial domains for the underlying scenario distribution to determine representative scenario clusters. Also, recall that the sizes of the two-stage bounding problems depend on the number of extreme points of the chosen simplicial domains. When extended to multi-stage problems, the simplicial scenario approximation can be applied either on the outcome domains at each time stage (referred to as *outcome approximation*) or on the joint scenario domain spanning the entire decision horizon (herein defined as *scenario approximation.*) The former approach constructs simplicial scenario trees based on the approximate outcomes generated at each time stage (see Frauendorfer [34] [35], for example.) The size of the resultant bounding problems using such an outcome approximation is determined by the product of the numbers of random variables at all time stages.

In contrast, the size of the two-stage bounding problems using a scenario approximation is determined by the sum of the numbers of random variables in all time stages, which grows much slower than the product, as the numbers of time stages and random variables increase. In this chapter, we propose a new scenario approximation approach to construct bounding problems for multi-stage models.

5.1 Formulation

We focus on multistage models in which the uncertainty space is discrete. Although the scenario approximation approach is applicable to both continuous and discrete cases, for computational efficiency - computing moment information without performing multidimensional numerical integration, we assume the underlying probability space is discrete. In the event that the underlying probability space is continuous, we can always approximate it by a discrete one with a large number of atoms.

The multi-stage model under consideration here is GLP_T in (2.16) on which we make the following additional assumption:

Assumption 5.1 *Technology matrices B_t, $t = 2, \ldots T$, are fixed (i.e., deterministic) but may vary from stage to stage.*

The assumption of fixed technology matrices is realistic for a rather broad class of applications including production planning and inventory control, and various resource allocation problems.

The multi-stage model GLP_T defined in (2.16) for the case of fixed technology matrices,

will be referred to by the same label GLP_T and is given below:

$$
\begin{aligned}
(\text{GLP}_T) \qquad & \min_{x} \ \left\{c_1 x_1 + \textstyle\sum_{t=2}^{T} \sum_{s \in S} \ p_s \ c_t(\eta_t^s) \ x_t^s\right\} \qquad (5.1)\\
& \text{s.t.} \quad A_1 x_1 = b_1 \\
& \qquad B_t x_{t-1}^s + A_t x_t^s = b_t(\xi_t^s) \ \ s \in S, \ t = 2, \ldots T \\
& \qquad x_t^s \geq 0 \ \ s \in S, \ t = 2, \ldots T \\
& \qquad \text{Nonanticipativity constraints} : x_t^s = z_t^N, \\
& \qquad\qquad \forall s \in N \ s \in S, \ N \in \mathcal{N}_t, \ t = 2, \ldots T-1.
\end{aligned}
$$

The above formulation is the so-called scenario formulation in which uncertainty is described by a set of scenarios $\{\omega^s := (\xi^s, \eta^s) : s \in S\}$ and x_t^s denotes the decisions (a scenario solution) at time t under a certain scenario ω^s, $t = 2, \ldots T$. Refer to Section 2.2.2 and the scenario tree in Figure 2.3 for an explanation of the nonanticipativity constraints on x^s.

5.2 Time stage aggregation

Suppose we assume that information about all of the random variables becomes available in the second stage. Then, one can make all subsequent decisions with complete knowledge of the scenario (path) being followed. This is equivalent to relaxing the information arrival structure on random events. The resulting relaxed decision problem is given by (5.1) with the nonanticipativity constraints being dropped. This relaxed multi-stage decision model can be written in the form of a two-stage model GLP_2 as given in (2.11). The relaxed model will be referred to by the same label GLP_2 and is given below:

$$
\begin{aligned}
(\text{GLP}_2) \qquad & \min_{x_1, y^s} \ \{c_1 x_1 + \textstyle\sum_{s \in S} \ p_s \ \mathbf{q}(\eta^s) \ y^s\} \qquad (5.2)\\
& \text{s.t.} \quad A_1 x_1 = b_1 \\
& \qquad \mathbf{T} x_1 + \mathbf{W} y^s = \mathbf{h}(\xi^s) \quad s \in S \\
& \qquad x_1 \geq 0, \ y^s \geq 0 \qquad\qquad s \in S,
\end{aligned}
$$

where

$$y^s = \begin{pmatrix} x_2^s \\ \vdots \\ x_T^s \end{pmatrix}, \quad \mathbf{T} = \begin{pmatrix} B_2 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \quad \mathbf{W} = \begin{bmatrix} A_2 & & & 0 \\ B_3 & A_3 & & \\ & \ddots & \ddots & \\ 0 & & B_T & A_T \end{bmatrix},$$

$$\mathbf{q}(\eta) = \mathbf{q}_0 + \sum_{t=2}^{T} \sum_{l=1}^{L} \mathbf{q}_{t,l} \cdot \eta_{t,l}\,, \quad \mathbf{q}_0 = \begin{pmatrix} c_{2,0} \\ \vdots \\ c_{T,0} \end{pmatrix}, \quad \mathbf{q}_{t,l} = \begin{pmatrix} 0 \\ . \\ 0 \\ c_{t,l} \\ 0 \\ . \\ 0 \end{pmatrix},$$

$$\mathbf{h}(\xi) = \mathbf{h}_0 + \sum_{t=2}^{T} \sum_{k=1}^{K} \mathbf{h}_{t,k} \cdot \xi_{t,k}\,, \quad \mathbf{h}_0 = \begin{pmatrix} b_{2,0} \\ \vdots \\ b_{T,0} \end{pmatrix}, \quad \mathbf{h}_{t,k} = \begin{pmatrix} 0 \\ . \\ 0 \\ b_{t,k} \\ 0 \\ . \\ 0 \end{pmatrix},$$

where $K = \sum_{t=2}^{T} K_t$, $L = \sum_{t=2}^{T} L_t$, and $I = K + L + 1$. The relaxed version GLP_2 of the true multistage problem GLP_T problem will be referred to as the *two-stage relaxation* of GLP_T. The decision stage obtained by aggregating stage 2 to stage T of GLP_T will be referred to as the aggregated second stage. In the aggregated second stage, decisions are made with full knowledge of the outcomes of the random vectors $(\omega_2, \ldots \omega_T)$. In other words, in the two-stage relaxation model, all of the decisions except for the first stage decisions x_1 are made under perfect information about the random events. Clearly,

$$V(GLP_2) \leq V(GLP_T), \tag{5.3}$$

where $V(.)$ is defined as the optimal objective value of the problem considered. By adding nonanticipativity constraints to GLP_2, one obtains GLP_T. This is equivalent to adapting the decision sequence in GLP_2 to the information arrival structure $\{\mathcal{N}_t,\ t = 2, \ldots T\}$ defined in Section 2.2.2.

5.3 Bounds for the multi-stage model

First consider the two-stage relaxation model, GLP_2 in 5.2. Let Ω denote a $(K+L)$-dimensional simplex Ω covering all of the scenarios ω^s, $s \in S$, of the joint random vector $\omega = (\omega_2, \ldots, \omega_T)' \in \Re^{K+L}$.

The direct application of the second-order simplicial scenario approximation [19] to the 2-stage relaxation problem GLP_2 yields the lower bounding problem LBLP_2 in (5.5) and upper bounding problem UBLP_2 in (5.6) for GLP_2. That is,

$$V(\mathrm{LBLP}_2) \leq V(\mathrm{GLP}_2) \leq V(\mathrm{UBLP}_2), \text{ where} \tag{5.4}$$

$$\begin{aligned} \mathrm{LBLP}_2: \quad \min_{x_1, y^i} \quad & \left\{ cx_1 + \textstyle\sum_{i=1}^{I} \rho_i \, \mathbf{q}(v^i)\, y^i \right\} \\ \text{s.t.} \quad & Ax_1 = b_1 \\ & \mathbf{T}x_1 + \mathbf{W}y^i = \mathbf{h}(\tilde{\xi}^i) \quad i = 1, \ldots I \\ & x_1 \geq 0, \;\; y^i \geq 0 \; i = 1, \ldots I, \end{aligned} \tag{5.5}$$

$$\begin{aligned} \mathrm{UBLP}_2: \quad \min_{x_1, y^i} \quad & \left\{ cx_1 + \textstyle\sum_{i=1}^{I} \rho_i \, \mathbf{q}(\hat{\eta}^i)\, y^i \right\} \\ \text{s.t.} \quad & Ax_1 = b_1 \\ & \mathbf{T}x_1 + \mathbf{W}y^i = \mathbf{h}(u^i) \quad i = 1, \ldots I \\ & x_1 \geq 0, \;\; y^i \geq 0 \; i = 1, \ldots I. \end{aligned} \tag{5.6}$$

Refer to Section 4.1.1 for the notation of the second-order approximations.

Furthermore, by partitioning the joint domain Ω under a suitable simplicial decomposition scheme, one may apply the 2-stage bounds iteratively, as in Chapter 4. From (5.3) and (5.4), it is clear that

$$V(\mathrm{LBLP}_2) \leq V(\mathrm{GLP}_T). \tag{5.7}$$

Although LBLP_2 provides a valid lower bound for GLP_T, the successive refinement of LBLP_2 under simplicial partitioning of Ω will not converge to $\mathrm{V}(\mathrm{GLP}_T)$, except in the case

when V(GLP_T)=V(GLP_2). The goal of this Chapter is to develop valid lower and upper bounds on the true multistage problem GLP_T, that would converge to V(GLP_T) under a suitable refinement scheme.

5.3.1 Multistage lower bound

The approach here is to begin with the 2-stage lower bounding problem and to tighten it further to yield multistage lower bound that can be forced to converge to the true problem iteratively.

First, consider constructing a model by row-aggregation of the constraints involving y^s, $s \in S$, in GLP_2 (5.2). A total of I $(= K + L + 1)$ sets of constraints will be constructed by a row aggregation of the constraints in GLP_2. Define the weights

$$\mu_i^s = \frac{p_s \, \lambda_i^s}{\rho_i}, s \in S, i = 1, \ldots I, \tag{5.8}$$

where ρ_i and λ^s are as defined in (4.2) and (4.27). It is easy to verify that each of the weight set $\{\mu_i^s \mid s \in S\}$ is convex, for each $i = 1, \ldots I$. That is, $0 \leq \mu_i^s \leq 1$ and $\sum_{s \in S} \mu_i^s = 1, \forall i = 1, \ldots I$. Multiplying the constraints involving y^s, $s \in S$, in GLP_2 by the i-th set of the convex weights $\{\mu_i^s \; s \in S\}$, corresponding to the index s and adding up all the weighted constraints result in the i-th set of aggregated constraints:

$$\sum_{s \in S} \mu_i^s \mathbf{T} x_1 + \sum_{s \in S} \mu_i^s \mathbf{W} y^s = \sum_{s \in S} \mu_i^s \mathbf{h}(\xi^s) \quad i = 1, \ldots I. \tag{5.9}$$

By the unity property of the convex weights and the affine property of the $\mathbf{h}(.)$ function, the constraint set in (5.9) can be rewritten as:

$$\mathbf{T} x_1 + \mathbf{W} \sum_{s \in S} \mu_i^s y^s = \mathbf{h}(\sum_{s \in S} \mu_i^s \xi^s) \quad i = 1, \ldots I. \tag{5.10}$$

The argument in the affine function $\mathbf{h}(.)$ on the right-hand side of (5.10) is precisely the approximation scenario $\tilde{\xi}^i$ which is defined in (4.5). Replacing the scenario constraints $\mathbf{T}x_1 + \mathbf{W}y^s = \mathbf{h}(\xi^s)$, $s \in S$, in GLP_2 (5.2) with the I sets of aggregated constraints in (5.10), we have the following problem:

$$\begin{array}{lll} \text{LBLP}_{2A}: \ \min_{x_1,y^s} & \{c_1x_1 + \sum_{s\in S}\ p_s\ \mathbf{q}(\eta^s)\ y^s\} & \\ \text{s.t.} & A_1x_1 & = b_1 \\ & \mathbf{T}x_1 + \mathbf{W}\sum_{s\in S}\mu_i^s\ y^s & = \mathbf{h}(\tilde{\xi}^i) \quad i = 1,\ldots I \\ & x_1 \geq 0; \quad y^s \geq 0 \ \ s \in S. & \end{array} \tag{5.11}$$

By the fact that $\eta^s = \sum_{i=1}^{I} \lambda_i^s v^i$ and the affine property of $\mathbf{q}(.)$ function, the second term of the objective function in (5.11) can be rewritten as

$$\sum_{s\in S} p_s \sum_{i=1}^{I} \lambda_i^s \mathbf{q}(v^i)\ y^s = \sum_{i=1}^{I} \rho_i \mathbf{q}(v^i) \sum_{s\in S} \frac{p_s\lambda_i^s}{\rho_i} y^s. \tag{5.12}$$

Thus, the problem LBLP_{2A} in (5.11) can be rewritten as follows:

$$\begin{array}{lll} \text{LBLP}_{2A}: \ \min_{x_1,y^s} & \left\{c_1x_1 + \sum_{i=1}^{I}\ \rho_i\ \mathbf{q}(v^i)\ \sum_{s\in S}\frac{p_s\ \lambda_i^s}{\rho_i}\ y^s\right\} & \\ \text{s.t.} & A_1x_1 = b_1 & \\ & \mathbf{T}x_1 + \mathbf{W}\sum_{s\in S}\frac{p_s\ \lambda_i^s}{\rho_i}\ y^s = \mathbf{h}(\tilde{\xi}^i) \quad i = 1,\ldots I & \\ & x_1 \geq 0; \quad y^s \geq 0 \ \ s \in S. & \end{array} \tag{5.13}$$

LBLP_{2A} is precisely the LBLP_2 in (5.5) augmented with the additional constraints:

$$y^i = \sum_{s\in S} \mu_i^s\ y^s, \quad \forall i = 1,\ldots I. \tag{5.14}$$

Therefore, the value of LBLP_{2A} is an upper bound on the value of LBLP_2. Since LBLP_{2A} can be obtained by aggregating the constraints in GLP_2, the value of LBLP_{2A} also provides a lower bound on the value of GLP_2. The bound results are stated formally as:

Proposition 5.2 $V(LBLP_2) \leq V(LBLP_{2A}) \leq V(GLP_2)$.

As GLP_2 is GLP_T without nonanticipativity (NA) constraints, by imposing NA constraints on y^s in LBLP_{2A}, we obtain a lower bounding problem for GLP_T:

$$\text{LBLP}_T\text{:} \begin{cases} \text{LBLP}_2 \\ \oplus\ y^i = \sum_{s\in S} \mu_i^s\, y^s, \quad i = 1,\ldots I \\ \oplus\ y^s \text{ nonanticipative } s \in S, \end{cases} \tag{5.15}$$

where $\oplus$ is defined as an operator which attaches a set of constraints to a problem. By rewriting the two constraint sets in (5.15), we present LBLP_T in an alternative form:

$$\text{LBLP}_T\text{:} \begin{cases} \text{LBLP}_2 \\ \oplus\ \textit{Aggregated} \text{ NA Constraints :} \\ \quad y_t^i = \sum_{N\in\mathcal{N}_t} \left(\sum_{s\in N} \mu_i^s\right) z_t^N, \ \ i = 1,\ldots I, \ \ t = 2,\ldots T-1, \\ \oplus\ T^{th} \text{ Stage Constraints :} \\ \quad y_T^i = \sum_{s\in S} \mu_i^s\, y_T^s, \ \ i = 1,\ldots I, \end{cases} \tag{5.16}$$

where the *aggregated NA constraints* in (5.16) are obtained by combining the nonanticipativity constrains:

$$y_t^s = z_t^N, \ \ \forall s \in N, \ \ s \in S, \ \ N \in \mathcal{N}_t, \ \ t = 2,\ldots T-1, \tag{5.17}$$

with the aggregation constraints in (5.14), and T^{th} stage constraints are just the aggregation constraints in (5.14) for the T^{th} stage.

Remarks. Under the assumption of "complete recourse" (refer to Chapter 2 for the definition), there exists a feasible solution for GLP_T, which implies that LBLP_T is feasible. Obviously,

$$V(LBLP_{2A}) \le V(LBLP_T) \le V(GLP_T). \tag{5.18}$$

As mentioned above, the T^{th} stage constraints in (5.16) are pure aggregation constraints and do not involve any NA constraints, thus, are not expected to add much value to the formulation in (5.16). On the other hand, by dropping these T^{th} stage constraints, we

expect to save significant computational effort since adding these constraints involves a large number (the cardinality of $\mathcal{N}_T = |S|$) of variables. In the remainder of this chapter, LBLP_T refers to the formulation in (5.16) without the T^{th} stage constraints.

5.3.2 Multistage upper bound

A simple heuristic for constructing an upper bound for the multistage problem GLP_T in (5.1) is to find a feasible solution of GLP_T. The following methodology is an effort in this direction.

We first show that a feasible solution of the 2-stage relaxation problem GLP_2 (5.2) can be constructed from a feasible solution of the 2-stage upper bounding problem UBLP_2 (5.6). Consider the constraints:

$$\mathbf{T}x_1 + \mathbf{W}\sum_{i=1}^{I} \lambda_i^s\, y^i = \sum_{i=1}^{I} \lambda_i^s\, \mathbf{h}(u^i) \;\; = \mathbf{h}(\xi^s), \quad s \in S, \tag{5.19}$$

which are obtained by aggregating the constraints involving variables y^i in UBLP_2 (5.6) by using the weights $(\lambda_i^s,\, i = 1, \ldots I)$, $s \in S$. Any feasible solution of the 2-stage upper bounding problem UBLP_2, denoted $(\hat{x}_1; \hat{y}^i,\, i = 1, \ldots I)$, will satisfy the aggregated constraints in (5.19). It develops that

$$\tilde{y}^s = \sum_{i=1}^{I} \lambda_i^s\, \hat{y}^i, \quad s \in S \tag{5.20}$$

is a feasible solution to the 2-stage relaxation problem GLP_2 in (5.2).

To ensure that the constructed scenario solution $(\tilde{y}^s,\; s \in S)$ is feasible and implementable to the true multistage problem GLP_T in (5.1), we only have to impose the nonanticipativity constraints on the constructed scenarios solution. That is, require that $\tilde{y}_t^s = \tilde{y}_t^{s'}$, for all $s \neq s'$ such that $s, s' \in N$, for all $N \in \mathcal{N}_t$, $\; t = 2, \ldots T-1$. This leads to the following

multistage upper bounding problem:

$$\text{UBLP}_T\text{:} \begin{cases} \text{UBLP}_2 \\ \oplus \textit{ Induced } \text{NA Constraints} : \sum_{i=1}^{I} \lambda_i^s \, y_t^i = \sum_{i=1}^{I} \lambda_i^{s'} \, y_t^i \\ \qquad \forall s \neq s', \;\; s, s' \in N, \;\; N \in \mathcal{N}_t, \;\; t = 2, \ldots T-1. \end{cases} \tag{5.21}$$

The added constraint set in (5.21) is referred to as the *induced* nonanticipativity constraints in order to distinguish it from the original nonanticipativity constraints. The induced nonanticipativity constraints (on the constructed solutions) can be expressed equivalently as follows:

$$\sum_{i=1}^{I} \lambda_i^s \, y_t^i = z_t^N, \quad \forall s \in N, \quad N \in \mathcal{N}_t, \quad t = 2, \ldots T-1. \tag{5.22}$$

Proposition 5.3 *$UBLP_T$ provides an upper bound for GLP_T.*

Proof. If UBLP_T is not feasible, by convention $\text{V}(\text{UBLP}_T)=\infty$, which is trivially an upper bound for GLP_T, but not a usable upper bound. If UBLP_T is feasible, the objective value attained by any feasible solution $(\hat{x}_1;\ \hat{y}^i,\ i = 1, \ldots I)$ is

$$\begin{aligned} & c_1 \hat{x}_1 \ + \ \sum_{i=1}^{I} \rho_i \, \mathbf{q}(\hat{\eta}^i) \, \hat{y}^i \\ = \ & c_1 \hat{x}_1 \ + \ \sum_{i=1}^{I} \rho_i \left(\sum_{s \in S} \frac{p_s \, \lambda_i^s}{\rho_i} \, \mathbf{q}(\eta^s) \right) \hat{y}^i \\ = \ & c_1 \hat{x}_1 \ + \ \sum_{i=1}^{I} \left(\sum_{s \in S} p_s \, \mathbf{q}(\eta^s) \, \lambda_i^s \right) \hat{y}^i \\ = \ & c_1 \hat{x}_1 \ + \ \sum_{s \in S} p_s \, \mathbf{q}(\eta^s) \sum_{i=1}^{I} \lambda_i^s \, \hat{y}^i \\ = \ & c_1 \hat{x}_1 \ + \ \sum_{s \in S} p_s \, \mathbf{q}(\eta^s) \, \tilde{y}^s \\ \geq \ & V(GLP_T). \end{aligned}$$

It is apparent that the best choice among all feasible solutions of UBLP_T for constructing a feasible solution of GLP_T is the optimal solution of UBLP_T, which gives the smallest

objective value for UBLP$_T$ and thus gives the smallest upper bound for GLP$_T$. That is,

$$V(UBLP_T) \geq V(GLP_T). \tag{5.23}$$

5.4 Refinement of the multistage bounds

The preceding basic multistage bounds will be improved by partitioning the joint scenario domain further. The basic refining practice here is similar to those for other bounds-based approximations, refer to [25] and Chapter 4, and it requires applying the bounding techniques on subsets of the joint scenario domains.

Since the derived multistage bounds utilize the second-order simplicial approximation, an initial ($K + L$ dimensional) simplex (Ω) containing the given set of scenarios is still needed, and when the domain Ω is partitioned, the ensuing sub-cells of the partition are also required to be simplices. The procedure of performing such a simplicial partition is described in Chapter 4. Some of the notation is restated in the following.

Consider a partition $\mathcal{P} := \{\Omega_r \mid r = 1, \ldots, R\}$ which partitions the joint simplicial domain Ω into mutually exclusive subsimplices Ω_r. Let the corresponding index-sets of scenarios be $S_1, \ldots, S_R$ where $S_r := \{s \mid \omega^s \in \Omega_r,\ s \in S\}$. The corresponding *cell probabilities* are evaluated as $P_r := \sum_{s \in S_r} p_s$ where $\sum_{r=1}^{R} P_r = 1$.

Computed according to a procedure similar to (4.1)–(4.6) using the cell-conditional first and second moments, the upper and lower approximating scenario distributions for cell Ω_r are denoted by $\{(\hat{\omega}^i(r), \rho_i(r)) \mid i = 1, \ldots, I\}$ and $\{(\tilde{\omega}^i(r), \rho_i(r)) \mid i = 1, \ldots, I\}$, respectively, where $\hat{\omega}^i(r) = (u^{i,r}, \hat{\eta}^{i,r})$, and $\tilde{\omega}^i(r) = (\tilde{\xi}^{i,r}, v^{i,r})$.

The improved lower and upper bounding models under the above-mentioned simplicial partition are stated below. The derivation is omitted as it is straightforward. The improved

lower bounding model is

$$\text{LBLP}_T(\mathcal{P}): \begin{cases} \text{LBLP}_2(\mathcal{P}) \\ \oplus \textit{Aggregated} \text{ NA Constraints}(\mathcal{P}): \\ \quad y_t^{i,r} = \sum_{N \in \mathcal{N}_t} \left(\sum_{s \in N, s \in S_r} \mu_{i,r}^s \right) z_t^N, \\ \quad r = 1, \ldots R, \;\; i = 1, \ldots I, \;\; t = 2, \ldots T-1, \end{cases} \tag{5.24}$$

where $\mu_{i,r}^s = \frac{p_s \lambda_{i,r}^s}{\rho_i(r)}$, $r = 1, \ldots R$, $i = 1, \ldots I$, $s \in S_r$, and $\text{LBLP}_2(\mathcal{P})$ is

$$\begin{aligned} \text{LBLP}_2(\mathcal{P}): \quad & \min_{x_1, y^{i,r}} \;\; \left\{ cx_1 + \textstyle\sum_{r=1}^R P_r \sum_{i=1}^I \rho_i(r)\, \mathbf{q}(v^{i,r})\, y^{i,r} \right\} \\ & \text{s.t.} \quad Ax_1 = b_1 \\ & \qquad \mathbf{T}x_1 + \mathbf{W}y^{i,r} = \mathbf{h}(\tilde{\xi}^{i,r}) \;\; r = 1, \ldots R, \;\; i = 1, \ldots I \\ & \qquad x_1 \geq 0; \quad y^{i,r} \geq 0 \;\; r = 1, \ldots R, \;\; i = 1, \ldots I, \end{aligned} \tag{5.25}$$

and the improved upper bounding model is

$$\text{UBLP}_T(\mathcal{P}): \begin{cases} \text{UBLP}_2(\mathcal{P}) \\ \oplus \textit{Induced} \text{ NA Constraints}(\mathcal{P}): \\ \quad \sum_{i=1}^I \lambda_{i,r}^s\, y_t^{i,r} = z_t^N \\ \quad \forall s \in N,\; s \in S_r,\; r = 1, \ldots R, \;\; N \in \mathcal{N}_t,\; t = 2, \ldots T-1, \end{cases} \tag{5.26}$$

where $\text{UBLP}_2(\mathcal{P})$ is

$$\begin{aligned} \text{UBLP}_2(\mathcal{P}): \quad & \min_{x_1, y^{i,r}} \;\; \left\{ cx_1 + \textstyle\sum_{i=1}^I P_r \left(\sum_{i=1}^I \rho_i(r)\, \mathbf{q}(\hat{\eta}^{i,r})\, y^{i,r} \right) \right\} \\ & \text{s.t.} \quad Ax_1 = b_1 \\ & \qquad \mathbf{T}x_1 + \mathbf{W}y^{i,r} = \mathbf{h}(u^{i,r}) \;\; r = 1, \ldots R, \; i = 1, \ldots I \\ & \qquad x_1 \geq 0; \quad y^{i,r} \geq 0 \;\; r = 1, \ldots R, \;\; i = 1, \ldots I. \end{aligned} \tag{5.27}$$

An important issue is whether, under a refinement scheme using the aforementioned partitioning on the simplicial joint domains, the resulting sequence of the multistage lower bounds is non-decreasing and that of the multistage upper bounds is also non-increasing, that is,

$$V(GLP_T) \geq V(LBLP_T(\mathcal{P})) \geq V(LBLP_T), \text{and} \tag{5.28}$$

$$V(GLP_T) \leq V(UBLP_T(\mathcal{P})) \leq V(UBLP_T) \tag{5.29}$$

Theoretical aspects of the monotonic issue is beyond the scope of this chapter. Instead, we shall show the required monotonic behavior of the multistage bounds in a computational setting. In that sense, the bound improvement technique described here is termed a "heuristic partitioning procedure" or "HPP."

5.5 Decomposition algorithms for multi-stage bounds

Recall that in the case of 2-stage bounds, the dual block angular structures in LBLP_2 and UBLP_2 permit the use of large-scale decomposition techniques such as the L-shaped method ([70]. However, when the aggregated NA constraints in (5.16) and the induced NA constraints in (5.21) are added to the 2-stage bounding problems, the exploitable block angular structure will be destroyed.

In this section, new decomposition-based algorithms are developed based on the underlying structures of $\text{LBLP}_T(\mathcal{P})$ and $\text{UBLP}_T(\mathcal{P})$ in an attempt to solve the multi-stage bounding models computationally efficiently.

5.5.1 Method for solving the lower bounding model

The method developed for solving $\text{LBLP}_T(\mathcal{P})$ is a variant of a cutting plane algorithm (CPA) involving only feasibility cuts. The algorithm is outlined below, derivation of which is rather self-explanatory.

CPA for $\text{LBLP}_T(\mathcal{P})$:

Step 1: Solve the master problem $\text{LBLP}_2(\mathcal{P})$ and obtain the optimal primal solution

$(\bar{x}_1;\ \bar{y}_t^{i,r},\ t = 2,\ldots T,\ r = 1,\ldots R,\ i = 1,\ldots I)$.

Step 2: Check the feasibility of $(\bar{y}_t^{i,r}, t = 2,\ldots T, r = 1,\ldots R, i = 1,\ldots I)$ in the aggregated nonanticipativity constraints by solving the following Phase-I feasibility subproblem:

$$\begin{aligned} \bar{w} = \quad & \min_{u,v} \quad \sum_{r=1}^{R}\sum_{i=1}^{I}\sum_{t=2}^{T-1}\sum_{j=1}^{n_t}\left(u_{t,j}^{i,r} + v_{t,j}^{i,r}\right) \qquad (5.30) \\ & \text{s.t.} \quad \sum_{N\in\mathcal{N}_t}\left(\sum_{s\in N, s\in S_r}\frac{p_s\,\lambda_{i,r}^{s}}{\rho_i(r)}\right) z_{t,j}^{N} + u_{t,j}^{i,r} - v_{t,j}^{i,r} = \bar{y}_{t,j}^{i,r}, \\ & \qquad r = 1,\ldots R, \quad i = 1,\ldots I, \\ & \qquad t = 2,\ldots T-1, \quad j = 1,\ldots n_t, \\ & \quad u \geq 0, \quad v \geq 0 \\ & \quad z_t^N \geq 0, \quad N \in \mathcal{N}_t, \quad t = 2,\ldots T-1, \end{aligned}$$

where u and v are artificial variables corresponding to the Phase-I simplex method.

Step 3: If $\bar{w} = 0$, then stop. The solution $(\bar{x}_1;\ \bar{y}_t^{i,r},\ r = 1,\ldots R,\ i = 1,\ldots I,\ t = 2,\ldots T)$ is an optimal solution to $\text{LBLP}_T(\mathcal{P})$. Otherwise, generate a constraint (feasibility cut) of the form:

$$\sum_{r=1}^{R}\sum_{i=1}^{I}\sum_{t=2}^{T-1}\sum_{j=1}^{n_t}\bar{\sigma}_{t,j}^{i,r}\,y_{t,j}^{i,r} \leq 0, \qquad (5.31)$$

where $\bar{\sigma}$ is the dual solution vector of subproblem (5.30).

Step 4: Add the constraint to the master problem $\text{LBLP}_2(\mathcal{P})$ and solve the restricted master problem ($\text{LBLP}_2(\mathcal{P})$ with all the added constraints) to obtain a new solution $(\bar{x}_1;\ \bar{y}_t^{i,r},\ t = 2,\ldots T,\ r = 1,\ldots R,\ i = 1,\ldots I)$. Go to Step 2.

Note that the feasibility subproblem in (5.30) is separable in both t and j. A variant of the above algorithm, denoted by Decomposed CPA (DCPA), is to solve, in Step 2, a set of smaller Phase-I feasibility subproblems for each pair (t,j), $t = 2,\ldots T-1$, $j = 1,\ldots n_t$, and generate, in Step 3, a constraint (feasibility cut) $\sum_{r=1}^{R}\sum_{i=1}^{I}\bar{\sigma}_{t,j}^{i,r}y_{t,j}^{i,r}$ for each pair (t,j), $t = 2,\ldots T-1$, $j = 1,\ldots n_t$. Then, one can either add all of the constraints

generated in Step 3 to the master problem, referred to as a *multi-cut* strategy, or aggregate all the constraints generated in Step 3 to be one constraint as in (5.31) (by adding them together) and add it to the master problem, referred to as a *single-cut* strategy.

Solving LBLP_T using either the CPA or the DCPA not only avoids solving a larger linear program, but also allows the use of decomposition techniques in the initial Step 1. With constraints of the type in (5.31) added to LBLP_2, the restricted LBLP_2 can be solved efficiently by performing dual simplex pivot steps from the previous optimal basis.

5.5.2 Method for solving the upper bounding model

Notice that the $\text{UBLP}_T(\mathcal{P})$ has many induced NA constraints, and thus, a straightforward LP solution approach is computationally tedious, specially when there are many scenarios. In this section, an efficient solution procedure is developed based on relaxing the induced NA constraints within an augmented Lagrangian approach. This method dualizes the constraints (5.22) and imposes quadratic penalty in the objective function. Consider the following augmented Lagrangian objective function:

$$\min_{x_1, y^{i,r}} \Bigg\{ c_1 x_1 + \sum_{r=1}^{R} \sum_{i=1}^{I} \rho_i(r)\, q(\hat{\eta}^{i,r})\, y^{i,r} \qquad (5.32)$$
$$+ \sum_{r=1}^{R} \sum_{t=2}^{T-1} \sum_{N \in \mathcal{N}_t} \sum_{s \in N, s \in S_r} \left(\alpha_t^s \Big(\sum_{i=1}^{I} \lambda_{i,r}^s y_t^{i,r} - z_t^N \Big) + \frac{\gamma}{2} \left| \sum_{i=1}^{I} \lambda_{i,r}^s\, y_t^{i,r} - z_t^N \right|^2 \right) \Bigg\},$$

in place of the linear objective in $\text{UBLP}_2(\mathcal{P})$ in (5.27), where $(\alpha_t^s,\ s \in S,\ t = 2, \ldots T-1)$ is the vector of dual variables associated with the induced nonanticipativity constraints (5.22), and $\gamma > 0$ is a penalty parameter. The quadratic component provides the "proximal term" to ensure that the solution of the augmented Lagrangian converges to the true problem. Furthermore, for more efficient computation, separability of this problem by cell $r, r = 1, \ldots, R$, can be introduced, similar in approach to that in the Progressive Hedging

Algorithm (PHA) of Rockafellar and Wets [68]. This approach entails using estimates $\bar{z}^N$ and $\bar{\alpha}_t^s$ for the "linking" variables z_t^N and α_t^s. These estimates will be generated in the algorithmic steps given below. Let us assume first that a first stage solution x_1 is known. Then the upper bound computations are performed in the following manner:

PHA for UBLP$_T(\mathcal{P})$:

Step 0: Set the iteration counter $\nu = 0$. Initialize the penalty parameter $\gamma^\nu = 0$ and set the dual price vectors $\bar{\alpha}_t^s = 0$ for all $s \in S$.

Step 1: (Given a first-stage solution $\bar{x}_1$) The problem UBLP$_2(\mathcal{P})$ (5.27) is separable in the cells of the current partition. Solve the decomposed subproblems of UBLP$_2(\mathcal{P})$ for 2-stage solution $(\hat{y}_t^{i,r}(\nu),\ i = 1,\ldots I,\ t = 2,\ldots T)$ for each cell r, $r = 1,\ldots R$.

Step 2: Compute the estimate $\bar{z}$ by the probabilistically weighted sum:

$$\bar{z}_t^N(\nu) = \frac{\sum_{r=1}^{R} \sum_{s\in N, s\in S_r} p_s \left(\sum_{i=1}^{I} \lambda_{i,r}^s\, \hat{y}_t^{i,r}(\nu)\right)}{\left(\sum_{s\in N} p_s\right)}, \tag{5.33}$$

for all $N \in \mathcal{N}_t$, $t = 2,\ldots T-1$.

Step 3: Obtain multistage solution $\{\hat{y}_t^{i,r}(\nu+1),\ t = 2,\ldots T,\ i = 1,\ldots I\}$ for each cell r, $r = 1,\ldots R$, by solving the decomposed subproblems of UBLP$_2(\mathcal{P})$ with the given first-stage solution $\bar{x}_1$ and with the objective function of each subproblem being replaced by the estimated augmented Lagrangian objective function in (5.32) for each cell.

Step 4: Compute $\bar{z}_t^N(\nu+1)$ in the same way as (5.33) but substitute $\hat{y}_t^{i,r}(\nu+1)$ for $\hat{y}_t^{i,r}(\nu)$.

Step 5 Check the termination criterion (for given small $\epsilon > 0$)

$$\delta = \left\{ |\bar{z}(\nu) - \bar{z}(\nu+1)|^2 + \sum_{s\in S} p_s \,|\, \bar{y}^s - \bar{z}(\nu+1)|^2 \right\}^{\frac{1}{2}} < \epsilon, \tag{5.34}$$

where

$$\bar{y}^s = \sum_{i=1}^{I} \lambda_{i,r}^s \, \hat{y}_t^{i,r}(\nu+1), \;\; s \in S_r, \;\; r = 1, \ldots R. \tag{5.35}$$

If the termination criterion is satisfied, stop; otherwise, go to Step 6.

Step 6: Update the estimated dual multipliers by

$$\bar{\alpha}_t^s(\nu+1) \leftarrow \bar{\alpha}_t^s(\nu) + \gamma^\nu (\sum_{i=1}^{I} \lambda_i^s \hat{y}_t^{i,r}(\nu+1) - \bar{z}_t^N(\nu+1)) \tag{5.36}$$

for all $t = 2, \ldots T-1$, $s \in S_r$, $r = 1, \ldots R$. Adjust the penalty parameter $\gamma^\nu \leftarrow \gamma^{\nu+1}$ if desired. Set $\nu \leftarrow \nu + 1$ and go to Step 3.

5.6 Solution procedures and implementation

The implementation of the solution methodology for multistage stochastic programs involves utilizing and modifying the basic SMART modules and other supplementary modules described in section 4.3 and developed for the 2-stage case. These modules are extended to perform the tasks needed in the solution procedure for the multistage case described in the preceding sections. The module SOSA remains the same for performing second-order scenario approximation as partitioning is performed on the grand domain of scenarios. The modified modules include MBOUNDS for computing Multistage Bounds, and MSIMDEC (Simplicial Decomposition for Multistage case) which also incorporate the S-P procedure and C-R procedure described in section 4.3. The other modified supplementary modules are MSLP and MSCENGEN. The MSLP module consists of Kall-Keller-Mayer [49] GENSLPN code for generating 2-stage SLP's with modification for generating multistage SLP's. The MSCENGEN module generates multistage stochastic (scenario) data p_s and $\omega_t^s := (\xi_t^s, \eta_t^s), t = 2, \ldots T, s \in S$ and forms time stage nested scenarios ω^s.

Given a multistage stochastic linear program (MSLP) with its *core input data*: c_t, b_t, A_t, B_t, $t = 1,\ldots T$, and *stochastic data*: the scenario set $\{(\omega^s, p_s) \ : \ s \in S\}$, an initial simplex Ω enclosing the set of scenarios is constructed according the procedure described in section 4.2.1. The general solution procedure and the modules involved for solving MSLP are described in the following.

Step 1: Perform second-order simplicial scenario approximation by SOSA module.

Given the geometric and the clustering information of new created cells from the SIMDEC module, this SOSA module computes the required second-moment information for each new cell Ω_r and outputs the lower and upper approximating scenario distributions computed according to (4.1)-(4.6).

Step 2: Compute bounds by MBOUNDS module:

In this module, the fixed core input data and information from SOSA and SIMDEC are needed to form programming subproblems, and IBM's OSL solver is used to solve LP subproblems and "quadratic programming" subproblems to obtain bounds and solutions.

This module first generates the two-stage relaxation lower bounding problem LBLP_2 or $\text{LBLP}_2(\mathcal{P})$ in MPS format, and then the LP is solved to obtain the 2-stage lower bound, LB_2, and the 2-stage lower bounding x_1 solution, x_{L2}.

The next step is to obtain the multistage lower bound, LB_T, and the multistage lower bounding x_1 solution, x_{LT}. The program $\text{LBLP}_T(\mathcal{P})$ can be solved by the simplex method (in OSL) or solved by the CPA (DCPA) described in section 5.5.1. If the simplex method is selected, the module then adds the aggregated NA constraints to

the 2-stage model and then solves the appended problem. If the (decomposed) cutting plane algorithm is selected, then the (decomposed) feasibility subproblem (5.30) is generated in MPS format and solved. The CPA (DCPA) described in section (5.5.1) continues until the termination criterion is satisfied.

To obtain upper bounds, there are two options regarding which lower bounding x_1 solution to be used in forming the upper bounding problems:

1. X1_OPTION=A, if the 2-stage lower bounding x_1 solution, x_{L2}, is used,
2. X1_OPTION=C, if the multistage lower bounding x_1 solution, x_{LT} is used.

In addition, there are choices to use either the Simplex method or the PHA described in Section 5.5.2 to solve the multistage upper bounding programming problems.

If the Simplex method is selected, $\text{UBLP}_2(\mathcal{P})$ in (5.27) is formed in MPS format for the given lower bounding x_1 solution and is solved to obtain 2-stage upper bound, $\text{UB}_2(x_1)$. The induced NA constraints in (5.26) are then added to the 2-stage upper bounding problem, and the appended model is solved to obtain multistage upper bound, $\text{UB}_T(x_1)$.

If the PHA is selected, for the given lower bounding x_1 solution, the 2-stage upper bounding LPs for each cell Ω_r are formed in MPS format and are solved successively to obtain 2-stage cell upper bounds, $\text{UB}_2^r(x_1), r = 1, \ldots R$. The overall 2-stage upper bound can be obtained by accumulating the objective value for each cell and weighting by the associated probability $P_r, r = 1, \ldots R$, that is,

$$\text{UB}_2(x_1) = c_1 x_1 + \sum_{r=1}^{R} P_r \ \text{UB}_2^r(x_1) \tag{5.37}$$

To obtain multistage upper bound, $UB_T(x_1)$, the PHA described in Section 5.5.2 is invoked and continues until the termination criterion (5.34) is satisfied.

Subsequently, the MBOUND module checks if the maximum number of partitions allowed is exceeded or if the relative gap of the bounds is within the specified ε to determine if the refining procedure should be terminated. The relative gap is defined similarly to that in (4.21) with the following changes. The 2-stage bounds LB_2 and $UB_2(x_1)$ are used in computing the relative gap if X1_OPTION=A, whereas the multistgae bounds LB_T and $UB_T(x_1)$ will be used in computing the relative gap if X1_OPTION=C.

If none of the above criteria is satisfied, the lower bounding x_1 solution used and the weighted difference WDIFF of the corresponding upper/lower bounds for all the current cells become the input to the MSIMDEC module, and the procedure of refining bounds continues.

Step 3: Refine the current partition by MSIMDEC module.

This module is a slight modification of the SIMDEC module described in Section 4.3. The modification affects only the partitioning strategies and is described below.

1. Choose a cell Ω_r for further partitioning according to one of the following two rules:

   W - the maximum Weighted difference (WDIFF) of upper/lower bounds,

   S - the maximum number of scenarios.

   The WDIFF for each cell is the cell probability weighted difference between the cell upper bound and the cell lower bound. The current 2-stage cell bounds,

LB_2^r and $UB_2^r(x_{L2})$, are used in computing WDIFFs if X1_OPTION=A, whereas the current multistage cell bounds, LB_T^r and $UB_T^r(x_{L2})$, are used in computing WDIFFs if X1_OPTION=C.

2. Choose an edge (w^i, w^j) for partitioning according to one of the following two rules:

   D_1 - nonlinearity measure as defined in (4.41),

   D_2 - the longest edge.

3. Determine the partitioning point according to one of the following two rules:

   P_1 - the mean point, or precisely, the point intersected by the conditional-mean hyperplane described in (4.42),

   P_2: the mid-point $(w^i + w^j)/2$.

5.7 Computational results

The purpose of this section is to investigate by computation the validity of the proposed solution methodologies for the multistage stochastic model having fixed technology matrices. The convergent behavior of the multistage bounds as well as that of the first stage solutions x_1 will also be examined under a simplicial partitioning refinement scheme.

The proposed solution procedure described in the previous section was coded in FORTRAN77 and was run on UTK's SP2 supercomputer (IBM RS/6000). IBM's OSL solver was incorporated in the source code to solve LP subproblems and quadratic programming subproblems.

The code allows up to 5 time stages and 12 random variables per time stage (maximum $K_t = 6$ and maximum L_t=6, and up to 100 scenario outcomes per time stage but with a

Table 5.1: Categories of the randomly generated multistage test problems.

Problem category	T	$m_t \times n_t$ $t = 1, \ldots T$	Matrix density	(K_t, L_t) $t = 2, \ldots T$	# of Scen.	GLP_T size #rows by #columns
1	3	3×6	60%	(1,1)	20×20	1263×2526
2	3	3×6	60%	(1,1)	40×40	4923×9846
3	3	3×6	60%	(1,1)	60×60	10983×21966
4	3	3×6	60%	(2,0)	40×40	4923×9846
5	3	3×6	30%	(1,1)	40×40	4923×9846
6	4	3×6	10%	(1,1)	$10 \times 10 \times 10$	666×333

limit of 20000 scenarios totally. The maximum number of simplicial partitions allowed is 50. The core input has the limits: $m_1, n_1 \leq 40$ and $m_t, n_t \leq 20, t = 2, \ldots T$. The above limitations are set only due to memory restriction of the machine account.

The test problems used in the computations are randomly generated multi-stage stochastic linear programs which have fixed technology matrices. Only the complete-recourse matrix option is used in GENSLPN. Coefficients in the core input A_t, B_t, b_t are generated uniformly from the interval [-5,5], and coefficients in c_t are generated uniformly from the interval [0,5]. Scenario outcomes are generated such that the corresponding random variables have a maximum spread of [0,6].

Several categories of problems were generated for the purpose of testing. The information relevant to the sizes of the randomly generated test problems is provided in Table 5.1. Unless mentioned otherwise, the computational results presented here are based on the averaged result of at least three randomly generated test problems.

5.7.1 Solving $\text{LBLP}_T(\mathcal{P})$

We compare the performances of using the simplex method, the CPA, and DCPA described in Section 5.5.1 for solving $\text{LBLP}_T(\mathcal{P})$. Test problems from Category 1 to Category 3 were

used for this purpose. The CPA (DCPA) is terminated when one of the following criteria is satisfied:

1. The objective value of the feasibility subproblem, $\bar{w} = 0$ or $\bar{w} \leq 0.01\%$ of the maximum element of the RHSs in the feasibility subproblem (5.30).

2. The improvement of the multistage lower bound LB_T obtained by the CPA (DCPA) from iteration to iteration is too little, specifically, $(LB_T^{\nu+1} - LB_T^{\nu}) \leq 0.01\%$ of $\mathrm{LB}_T^{\nu+1}$, where ν is the iteration number in the CPA (DCPA).

Figure 5.1 shows the cumulative CPU time required for solving $\mathrm{LBLP}_T(\mathcal{P})$ by the three methods (up to only 10 partition iterations). The procedure based on the simplex method consistently requires less CPU time than the CPA and the DCPA in solving the lower bounding problems. The CPU time required by the CPA and the DCPA grows rapidly as the number of partitions increases, and it is almost intolerable. Therefore, the simplex method is preferred to the CPA and the DCPA and will be used to solve the multistage lower bounding problems for the remaining computations.

To complete the analyses, the objective values obtained by the CPA and the DCPA are reported here. For the case of 20x20 scenarios, the objective values obtained by the CPA (DCPA) range from 100% to 96.4% (100% to 96.7%) of the objective values obtained by the Simplex method. For the case of 40x40 scenarios, the objective values obtained by CPA range from 100% to 93.3%, and those obtained by DCPA range from 100% to 93.94%. For the case of 60x60 scenarios, the objective values obtained by CPA range from 99.99% to 92.54%, and those obtained by DCPA range from 99.25% to 93.45%. The reason why the CPA and the DCPA obtain different objective values could be due to degeneracy in dual

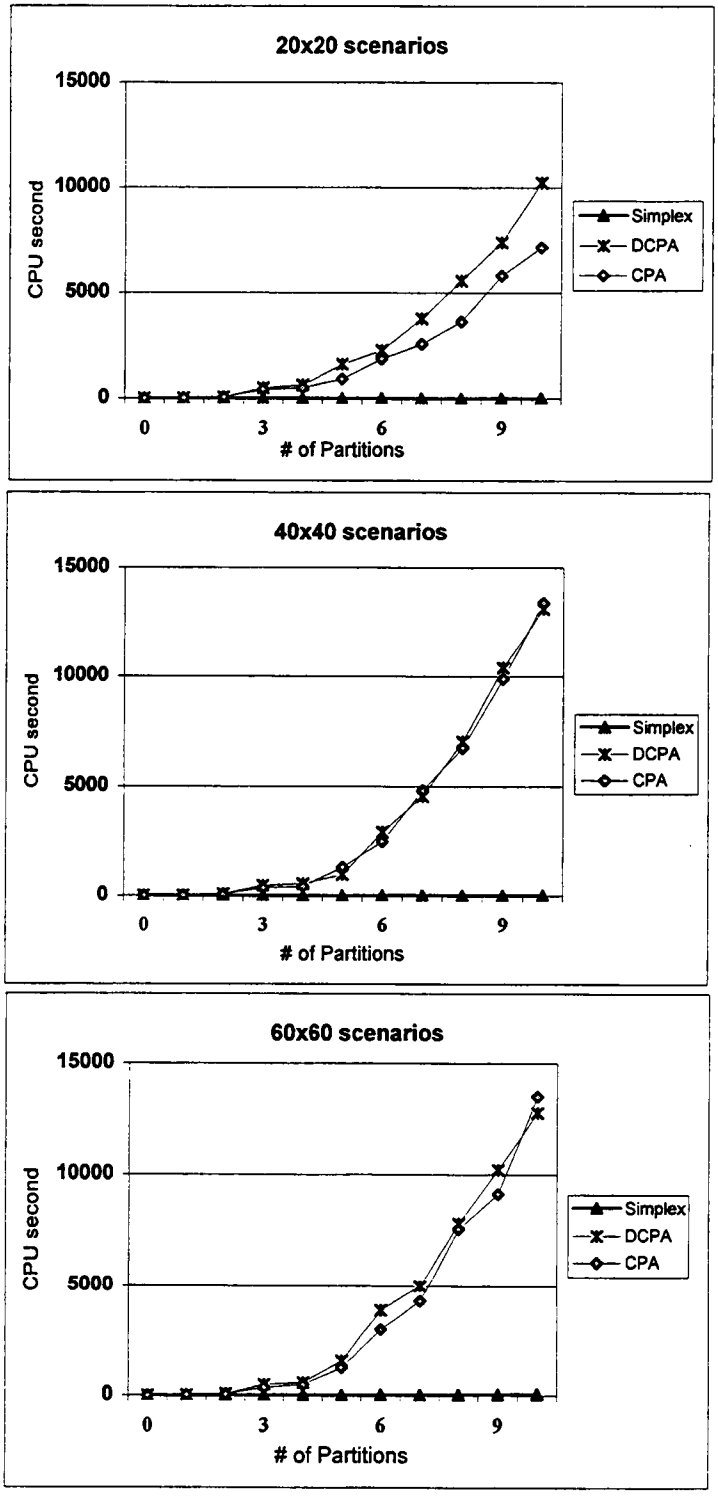


Figure 5.1: Cumulative CPU time for solving $\text{LBLP}_T(\mathcal{P})$

in the feasibility subproblem in (5.30). That is, the dual solutions obtained by the CPA and the DCPA are different, thus generating different feasibility cuts in (5.31) which then results in different LB_T.

5.7.2 Solving $UBLP_T(\mathcal{P})$

Test problems from Category 1 to Category 3 were solved to compare the performance of the two solution methods used for solving $UBLP_T(\mathcal{P})$. Each problem was solved by the two solution methods using four different partitioning strategies, and thus solving the problem in eight instances. In PHA, the fixed quadratic penalty parameter γ was fixed at .05 and the termination criterion is set for $\epsilon = .01$ for solving the upper bounding problems.

The cumulative average CPU time required for solving $UBLP_T(\mathcal{P})$ by the simplex method and by the PHA (up to only 20 partition iterations) are shown in Figure 5.2. The CPU time required for the simplex method grows linearly as number of partitions increases, while that for the PHA is exponential. With respect to the CPU time, the PHA has an advantage over simplex method when the problem size is moderate or large (see the case 60x60 scenarios in Figure 5.2).

Another reason why the PHA is preferred to simplex method for solving $UBLP_T(\mathcal{P})$ is that when large number of induced NA constraints are added to $UBLP_2(\mathcal{P})$, the problem become numerically unstable or could become nearly infeasible if solved by the simplex method. On the other hand, the PHA method is numerically more stable as evidenced by our computationally experience. To analyze the quality of the objective values obtained by PHA, the objective values obtained by PHA are represented as percentages of the objective values obtained by simplex method for those problems that are feasible or that have very

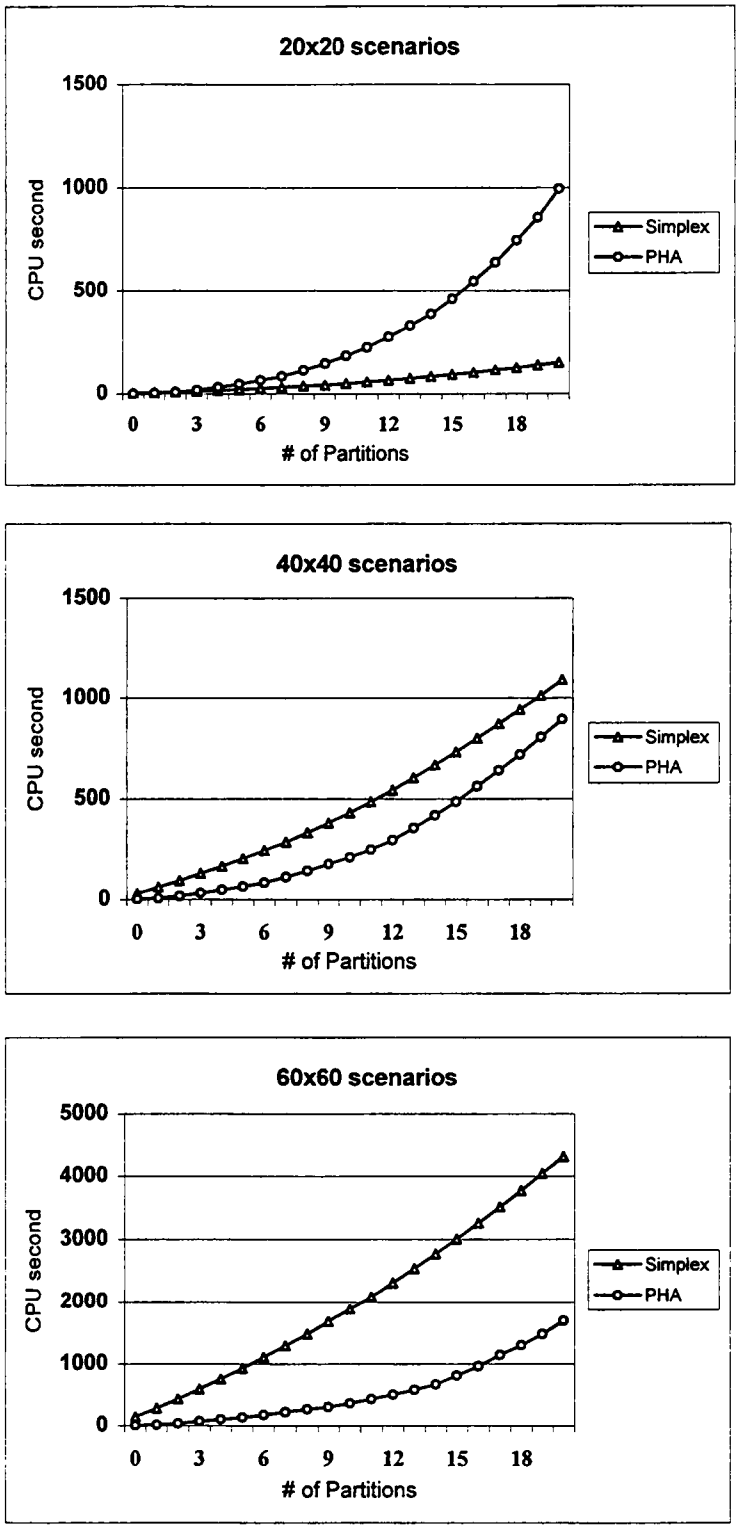


Figure 5.2: Cumulative CPU time for solving $\text{UBLP}_T(\mathcal{P})$

small amount of primal infeasibility (10^{-5}). The averaged objective values obtained by the PHA range from 99.37% to 100.09% of the objective values obtained by the simplex method with standard deviations 0.00% - 0.67%.

For all the following computations, PHA with fixed quadratic penalty parameter $\gamma = .05$ and termination criterion $\epsilon = .01$ was used to solve all the $\text{UBLP}_T(\mathcal{P})$.

5.7.3 Bound convergent behavior

We first verify that the multistage bounds behave monotonously under the the refinement scheme which partitions on the simplicial joint scenario domains. In some instances, the multistage optimal objective values are obtained at the initial iteration, that is $\text{LB}_T = \text{UB}_T$ without any partitioning. For the more general cases where partitioning is required, we shall show the behavior of the multistage bounds as the simplicial joint scenario domains are partitioned finer.

Figure 5.3 to Figure 5.9 show, for different test problems, how multistage bounds ($\text{UB}_T(x_L)$ and LB_T) improve toward their multistage optimal objective values within 20 partition iterations under one of the partitioning strategies. The behaviors of 2-stage bounds are also shown in Figure 5.3 to Figure 5.9 for the purpose of comparing with the multistage cases. The gaps between the 2-stage bounds and the multistage bounds can be attributed to the effect of the added NA constraints.

Generally, the multistage bounds behave similarly to their 2-stage counterpart, and the bounds improve monotonously as the joint scenario domains are partitioned finer. The improvement of the bounds occurs mostly in the first few partition iterations and tails off afterwards. The lower bound LB_T generally is closer to its optimal objective value than the

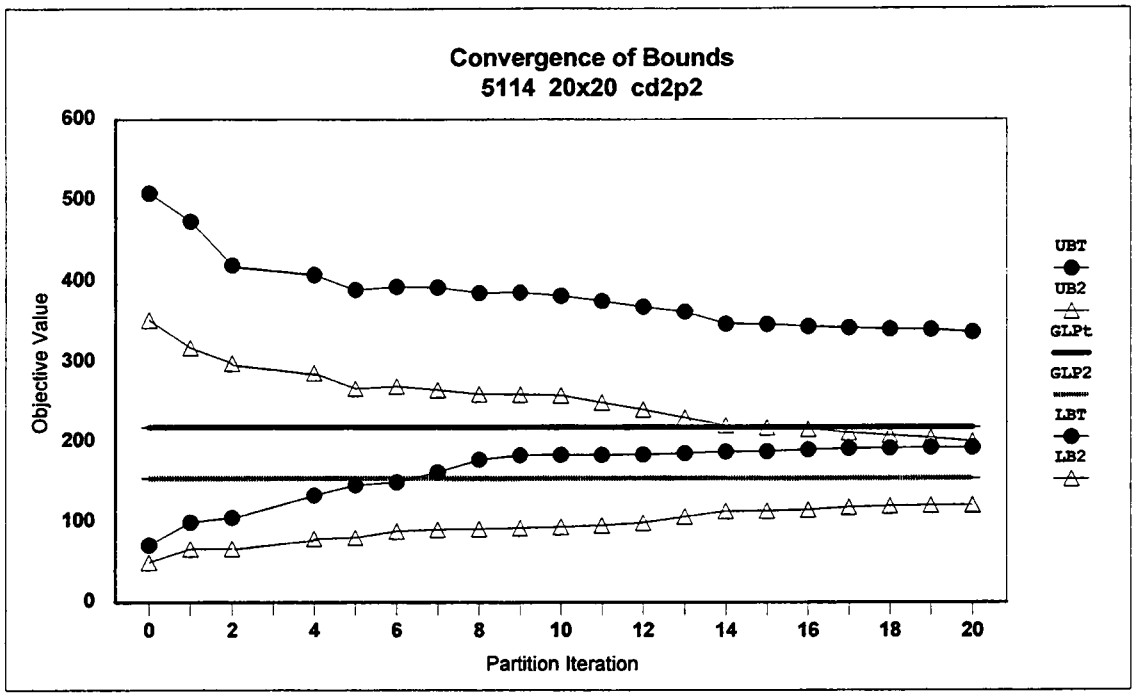


Figure 5.3: Bound behavior - a sample problem from Category 1.

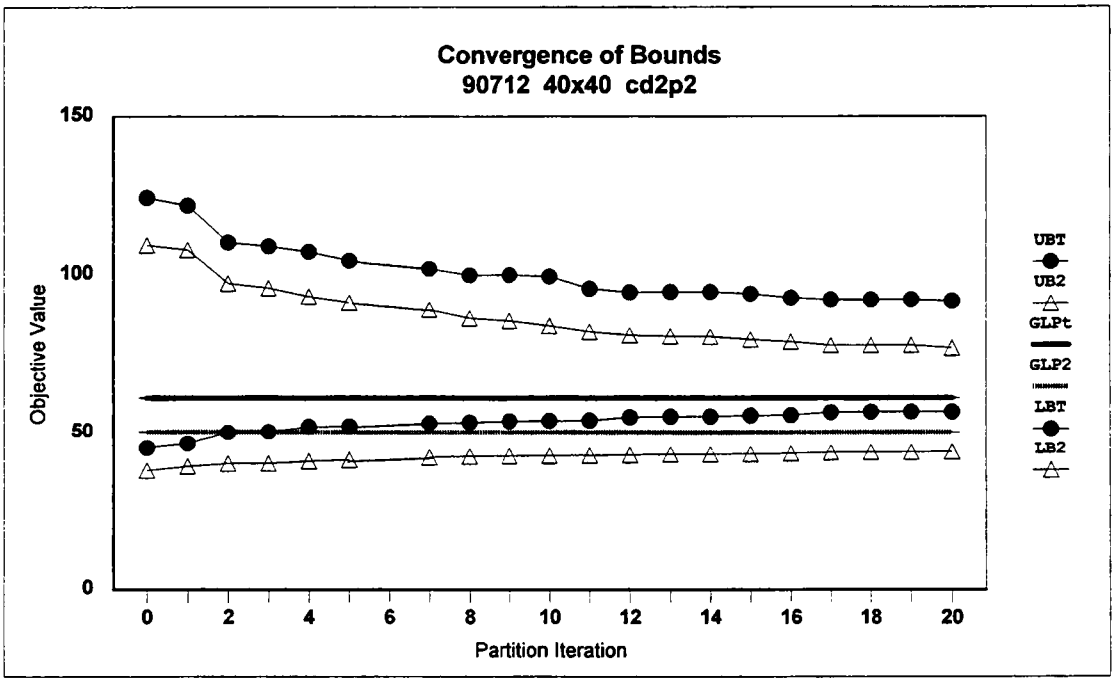


Figure 5.4: Bound behavior - a sample problem from Category 2.

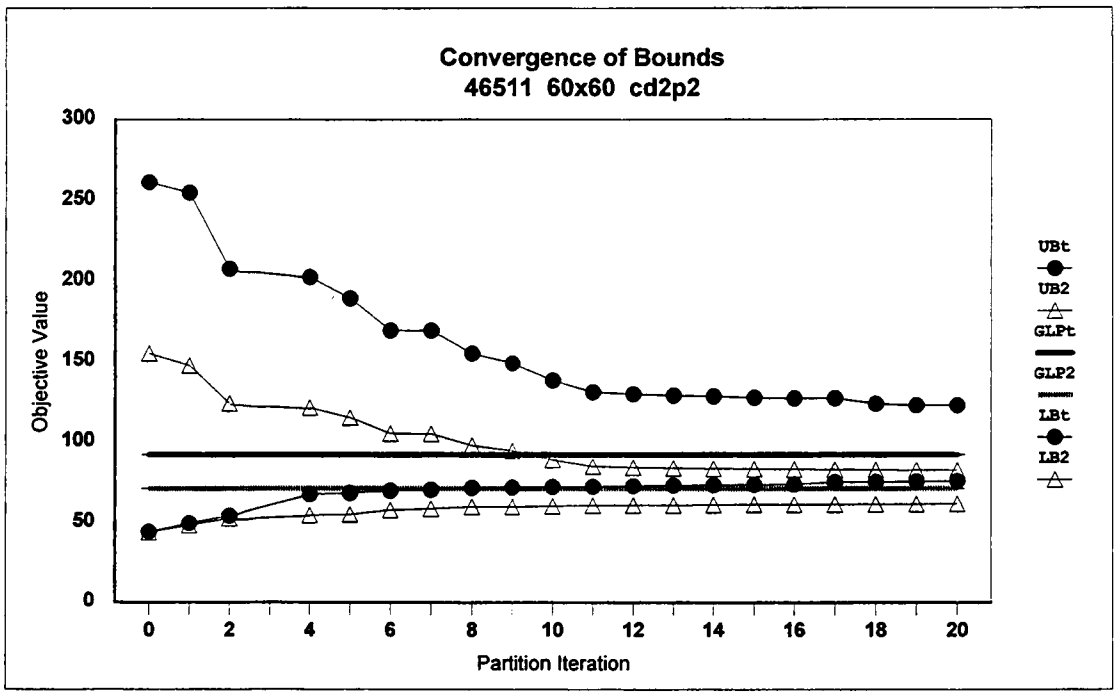


Figure 5.5: Bound behavior - a sample problem from Category 3.

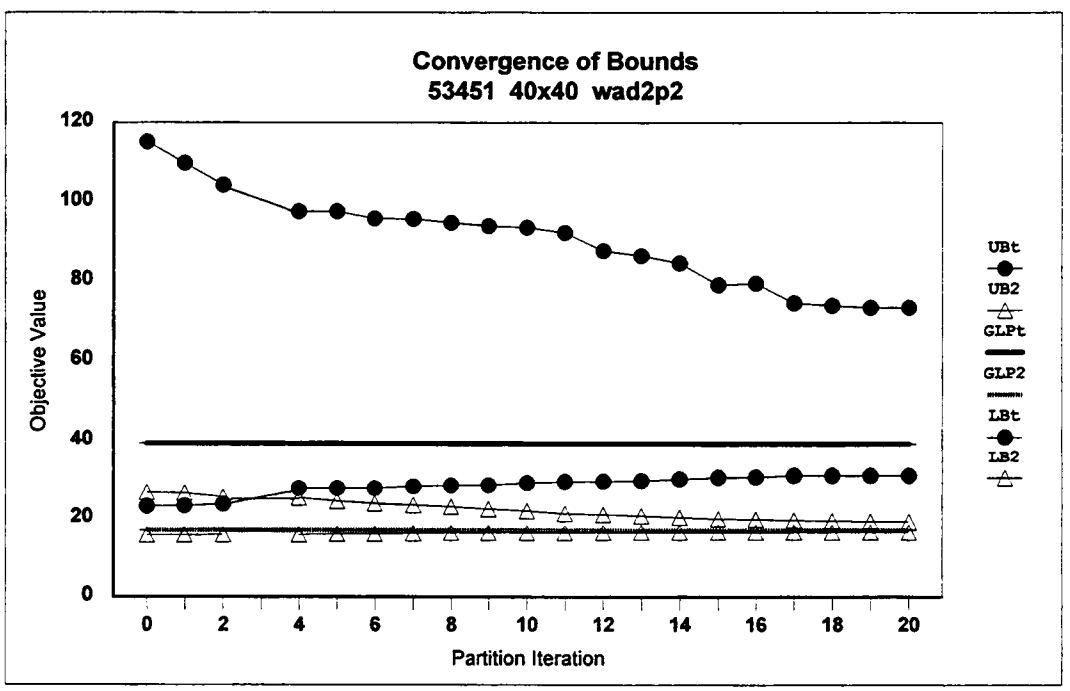


Figure 5.6: Bound behavior - a sample problem from Category 4.

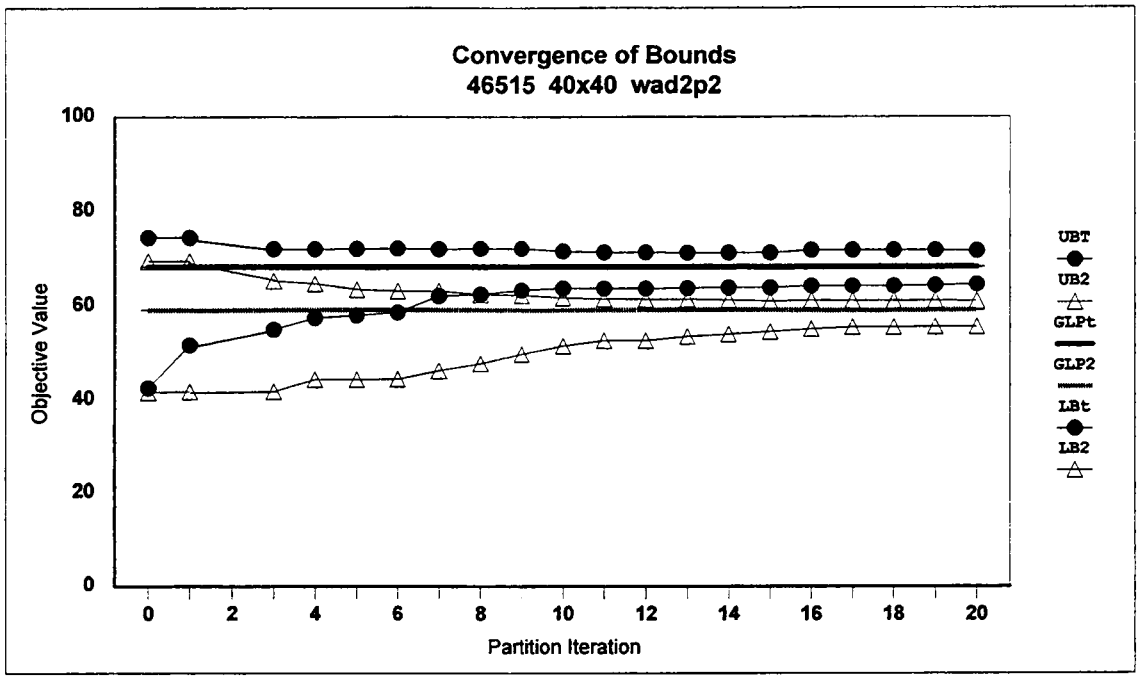


Figure 5.7: Bound behavior - a sample problem from Category 5.

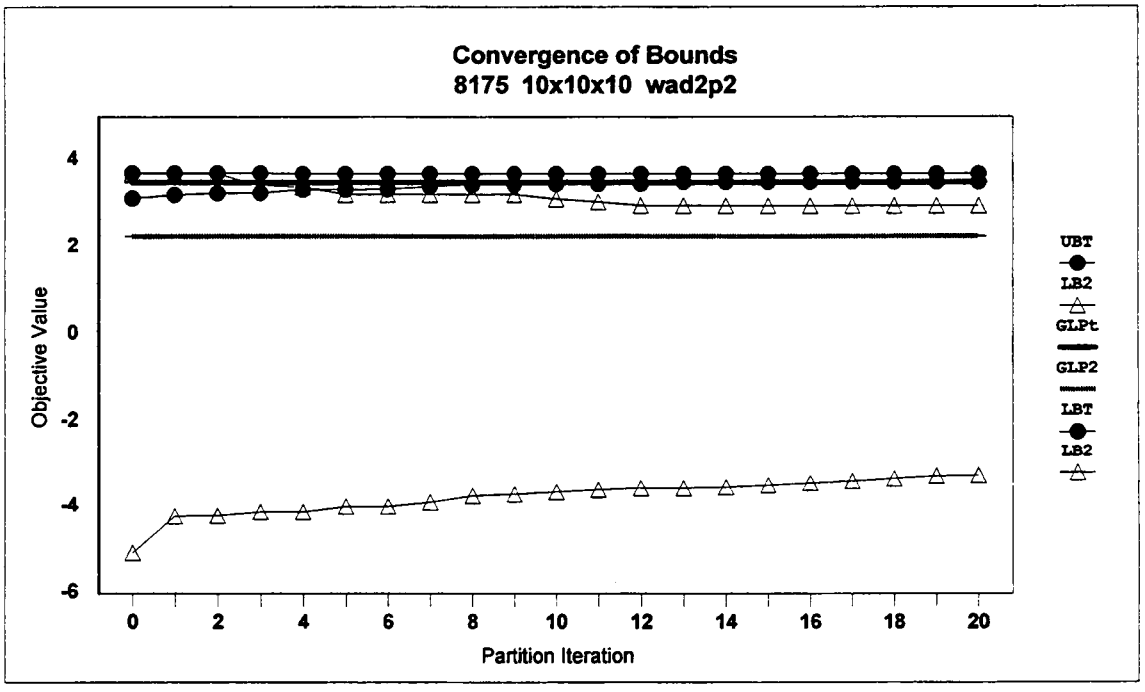


Figure 5.8: Bound behavior - a sample problem from Category 6.

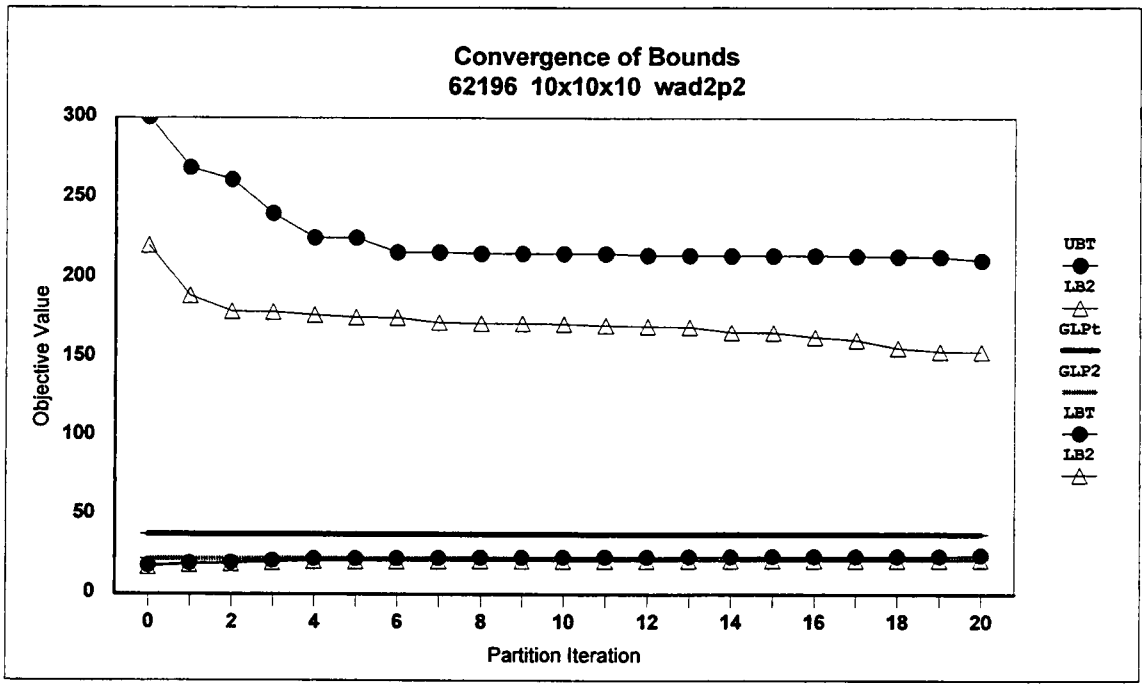


Figure 5.9: Bound behavior - a sample problem from Category 6.

upper bound.

Note that the upper bounds $\mathrm{UB}_T(x_L)$ reported here which used first-stage lower bounding solution x_L are weaker than the upper bounds UB_T which are obtained by solving $\mathrm{UBLP}_T(\mathcal{P})$ for its x_U. Therefore, $\mathrm{UB}_T(x_L)$ may not behave monotonously as the joint scenario domains are partitioned further. The motivation for using the upper bounds $\mathrm{UB}_T(x_L)$ is to obtain an easily computed upper bound for the purpose of picking partitioning cells and for the purpose of estabilshing termination criteria. Any valid upper bound can serve for these purposes.

A final note on the bound improving behaviors is that when the partition iteration increases, the bounds improve but the gaps between the 2-stage bounds and the multistage bounds remain about the same. The reason could be that as the joint scenario domains

are partitioned only the moment information of the scenarios is refined but the dynamic structure of the scenarios is not further refined.

In the remaining of the computation, we investigate some of the factors that may affect the magnitude and the speed the multistage bounds improve toward their optimal objective values.

The effect of partitioning strategies

Test problems from Category 2 were used for investigating the effect of the partitioning strategies on bound convergent behavior as well as on the convergent behavior of x_L. Each partitoning strategy is defined by the following components:

1. OPTION_X1: A or C,
2. cell choice: W or S,
3. edge choice: D1 or D2,
4. partitioning point: P1 or P2.

Refer to Section 5.6 for detailed description about each component of the partitioning strategies. The effect of partitioning strategies on the bound improving behavior of a test problem from Category 2 is depicted in Figure 5.10.

We further analyze the effect of each component of the partitioning strategies on the bound convergent behaviors and the convergent behavior of x_L. The convergent behaviors of the multistage cases as well as the 2-stage cases will both be presented for the purpose of comparison.

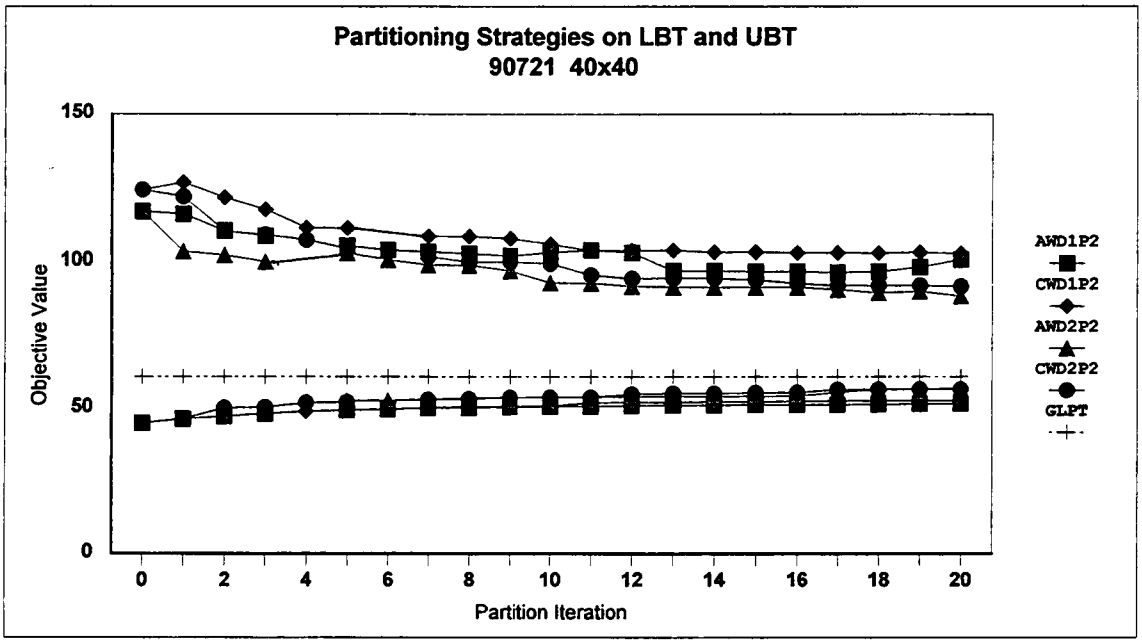


Figure 5.10: Effect of partitioning strategies on bound convergent behavior. An example using test problem from Category 2.

As shown in Figure 5.11 to Figure 5.14, none of the strategies has advantage over the other on the improvement of the multistage bounds. Some strategies appear to be better than the other initially, but the advantages diminish after about 10 partition iterations. The initial effects of the partitioning strategies and the diminishing effects are more noticeable in the 2-stage bound convergent behaviors. The improvement in bounds seems somewhat related to the pattern how first stage lower bounding solution converges to its optimal solution.

The effect of problem size

Test problems from Category 2 are used for investigating the effect of number of scenarios on bound improving behavior. Figure 5.15 shows the effect of scenario size on bound improving

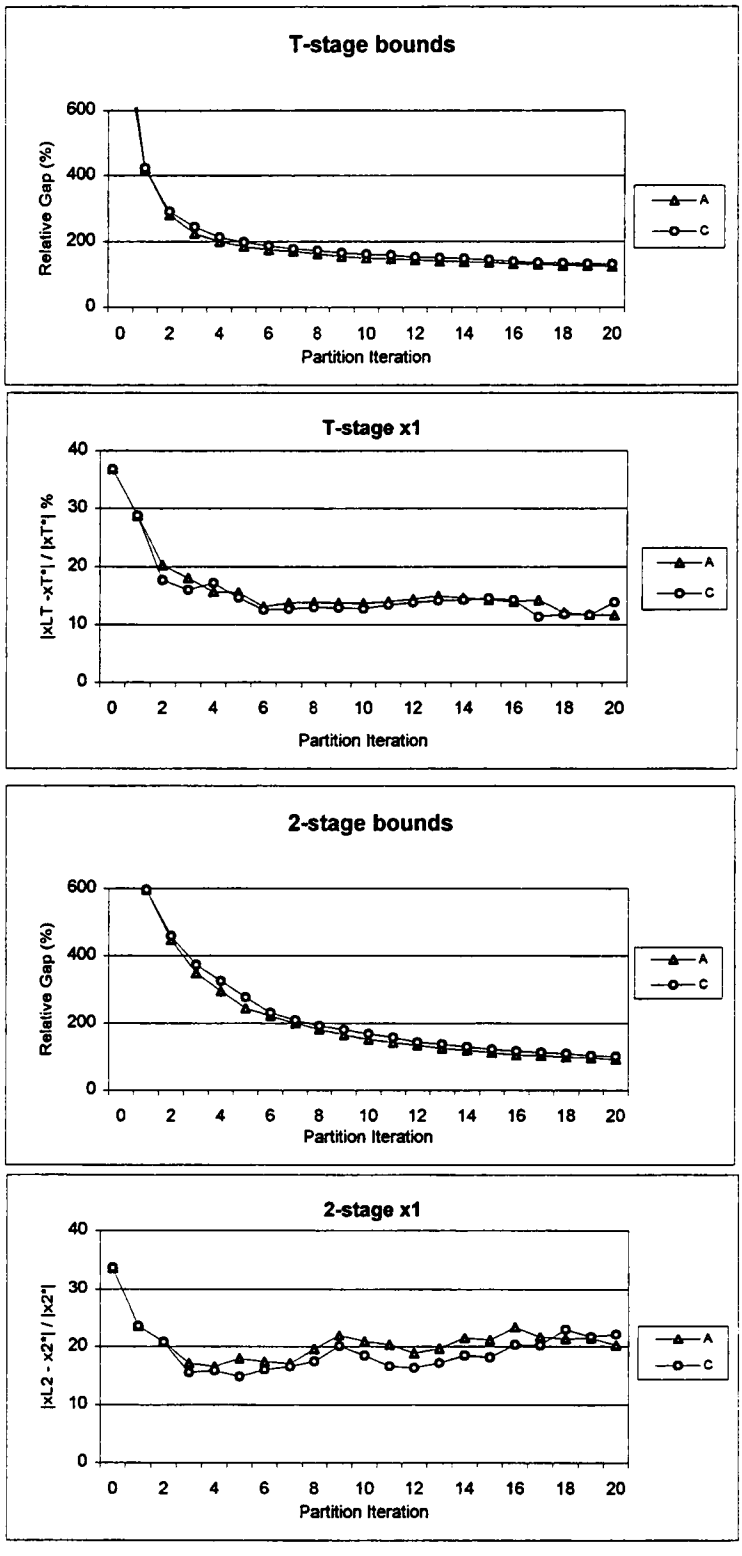


Figure 5.11: Effect of OPTION_X1 on bound convergent behavior.

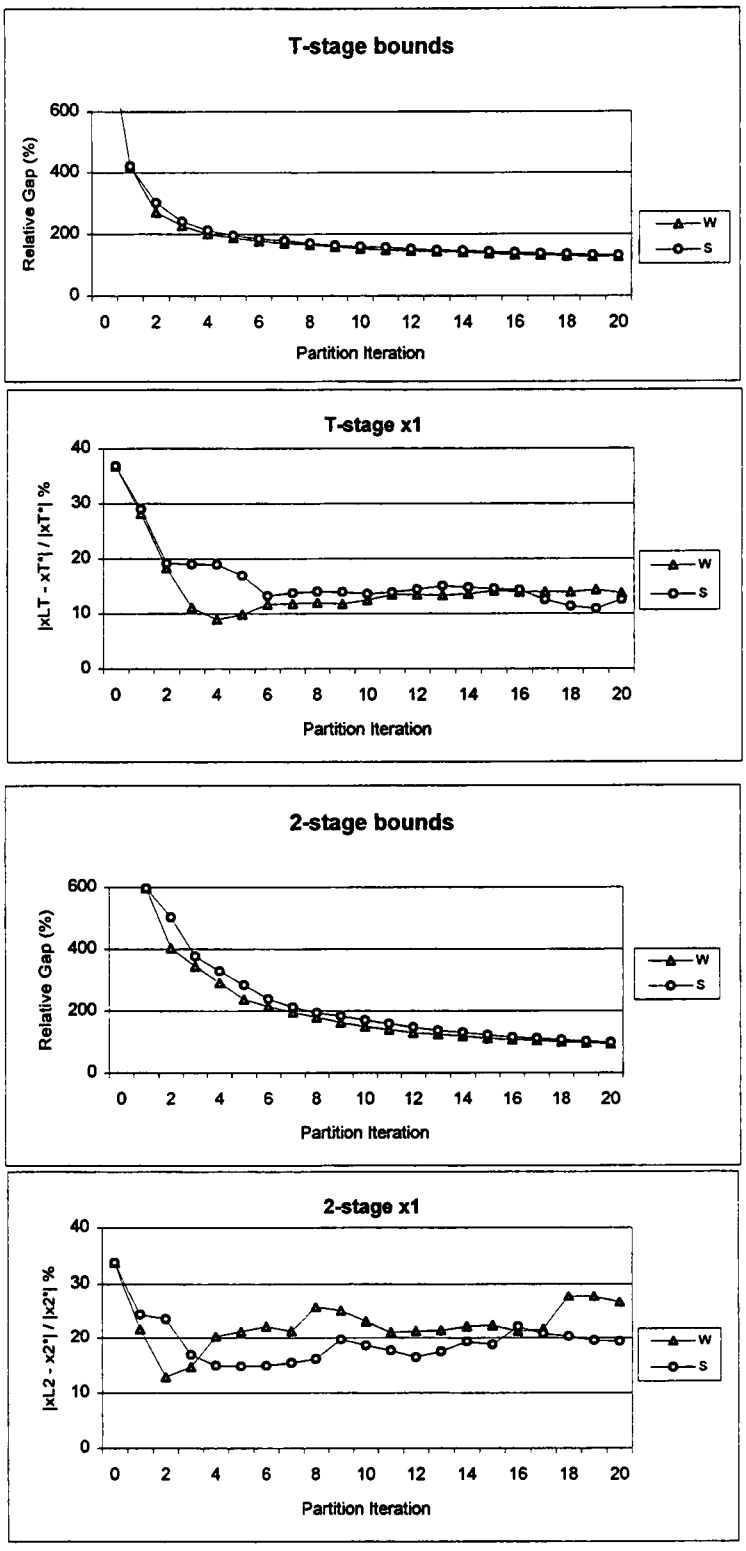


Figure 5.12: Effect of W: WDIFF vs S: Scenario partitioning on bound convergent behavior.

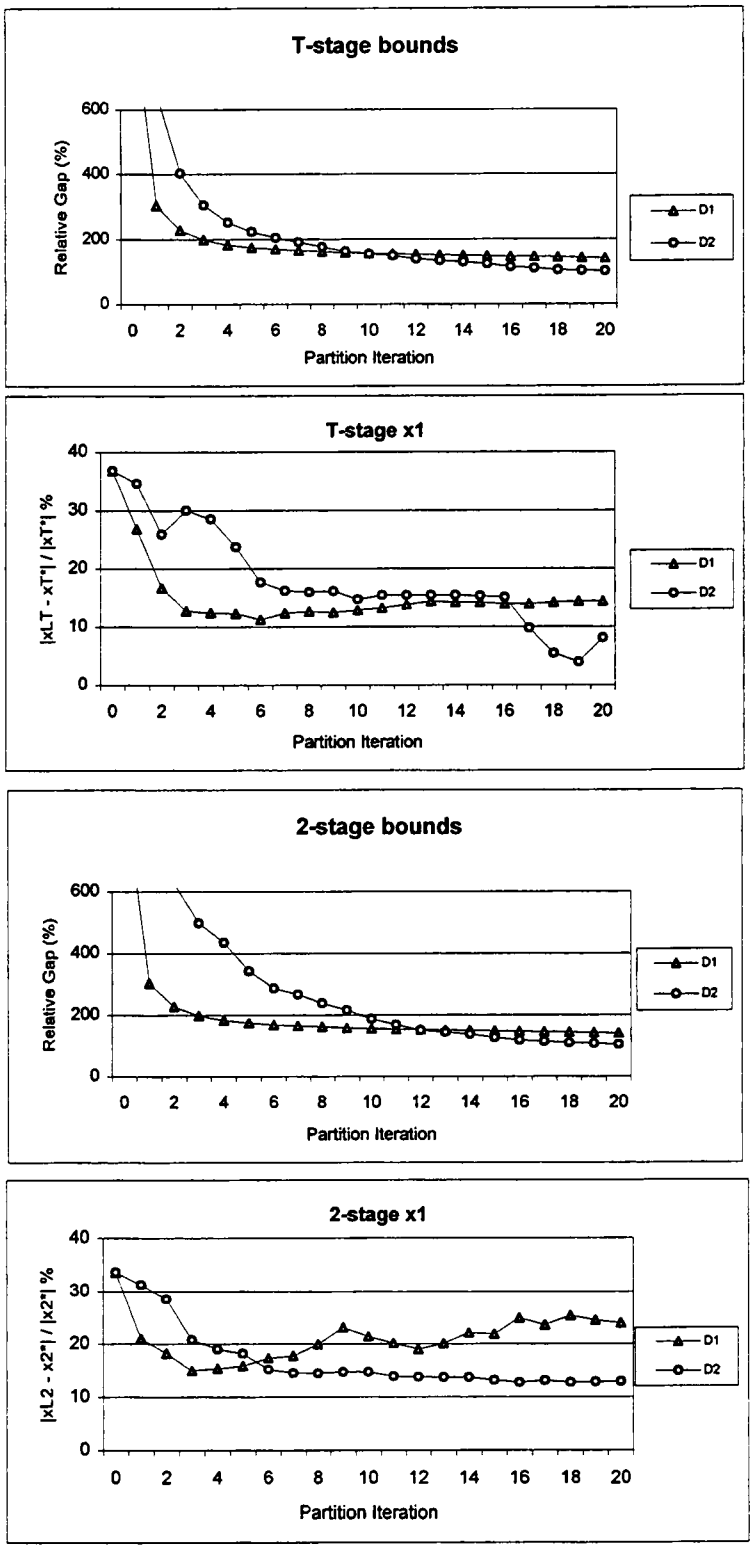


Figure 5.13: Effect of D1: Nonlinearity vs D2: Max length partitioning on bound convergent behavior.

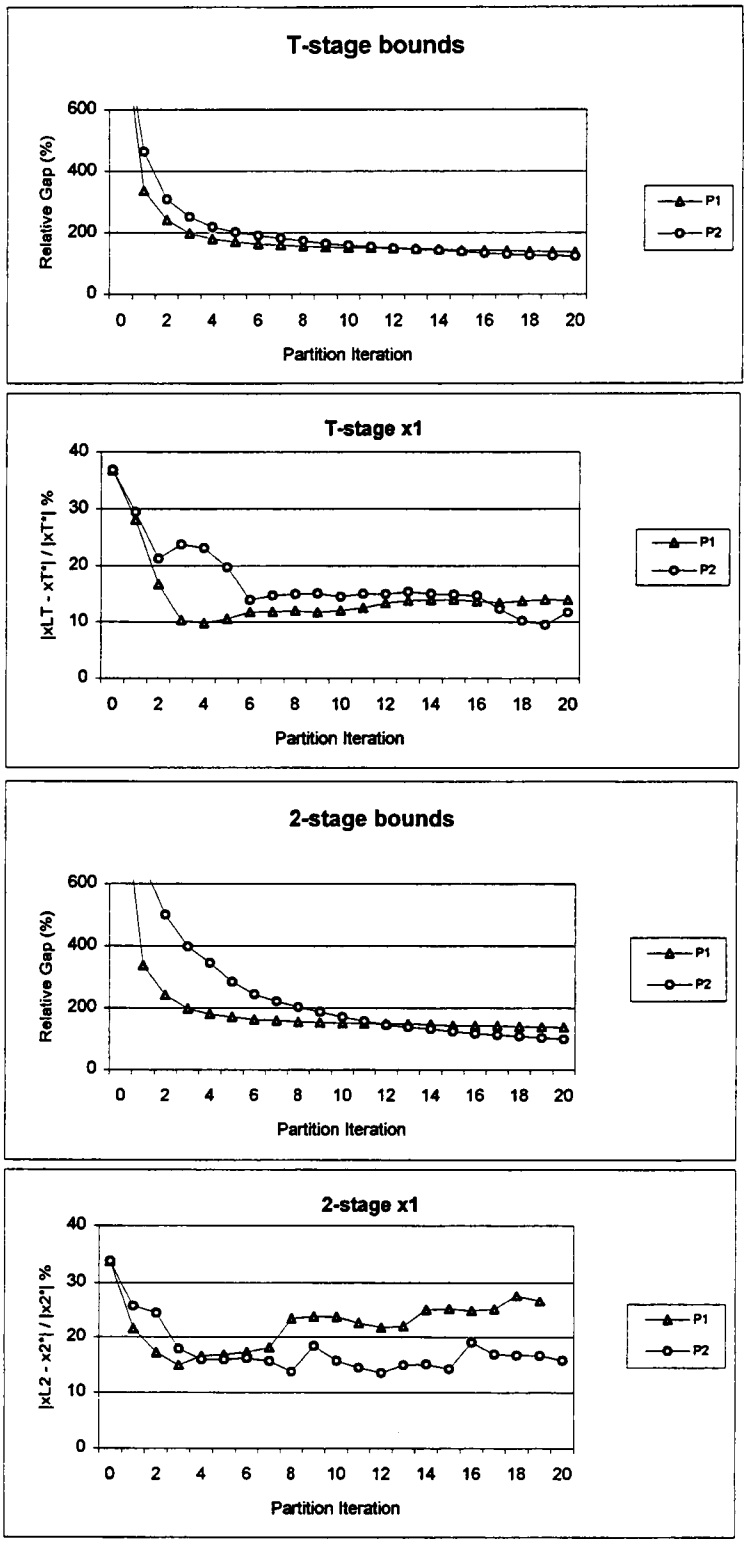


Figure 5.14: Effect of P1: Mean point vs P2: Mid point partitioning point on bound convergent behavior.

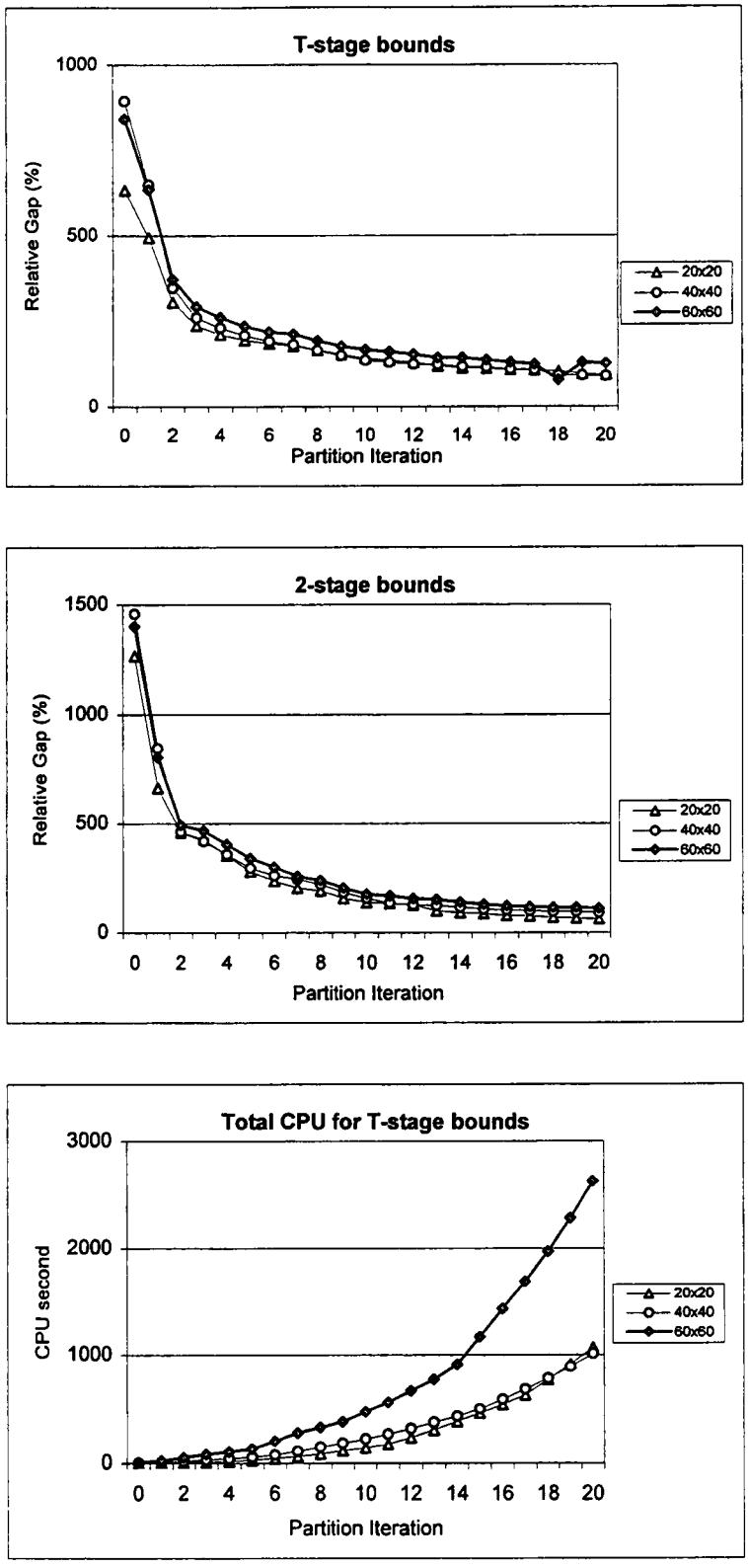


Figure 5.15: Effect of number of scenarios on bound convergent behavior.

behavior and on the total CPU time required to solve for multistage bounds. Generally, the rate of the bound improving is not affected by scenario size, but the relative bound gap appears to be larger when the size of scenario set is larger. The effect of scenario size does not affect the 2-stage case as much as the multistage case. The total CPU time required for solving the multistage bounds increases as the size of the scenario set increases.

The effect of problem structure

Test problems from Category 2 and Category 4 were used to compare the bound improving behaviors of saddle functions with that of convex functions. Figure 5.16 shows the comparison of the bound improvement and total CPU time required for saddle case and convex case. The initial relative gap of convex case appears to be smaller than that of saddle case. However, the bound improving rate of convex case is not greater than that of saddle case. The CPU time required also indicates that convex case doesn't seem to be easier to solve than saddle case.

Test problems from Category 2 and Category 5 were used to investigating the effect of matrix density on bound improving behaviors. Figure 5.17 show the comparison of the bound improvement and total CPU time required for problems with dense (60% density) matrices and problems with sparse (30% density) matrices. The initial bound relative gap and the bound improving rate seem to indicate that 30% density case is easier to solve than 60% density case. However, the total CPU time required seems to indicate the converse.

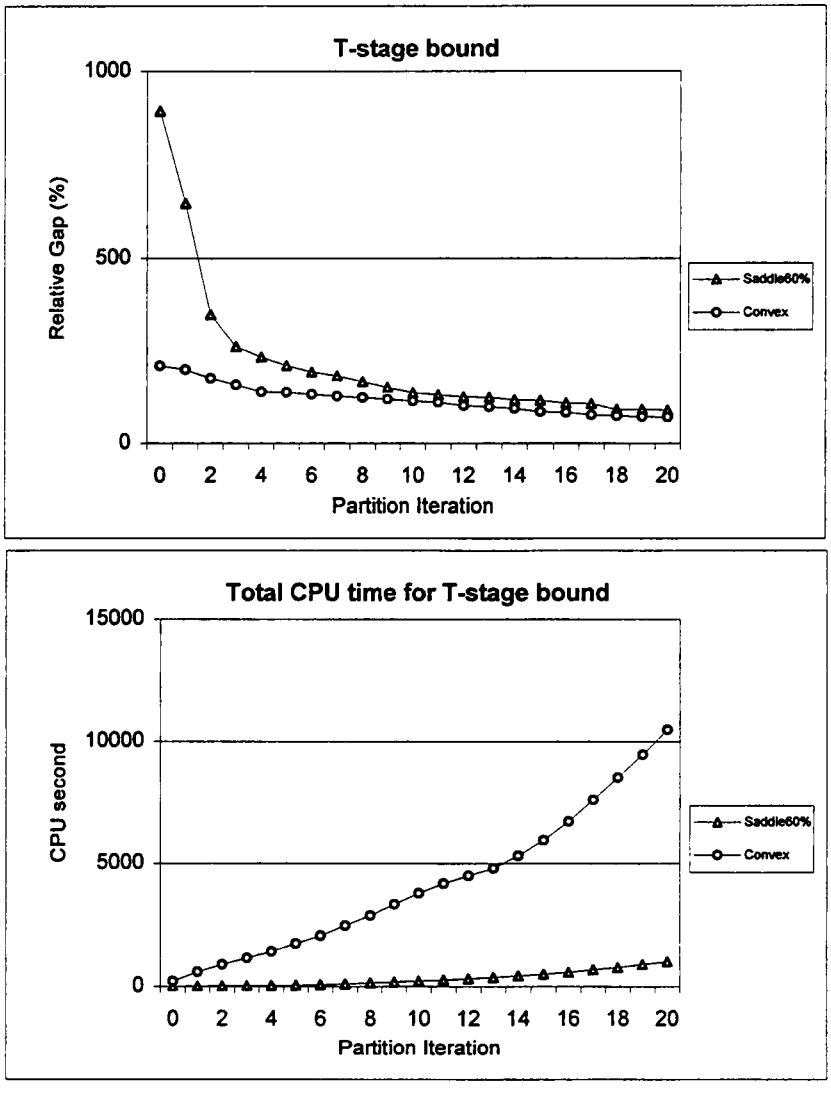


Figure 5.16: Effect of saddle function vs convex function on bound convergent behavior.

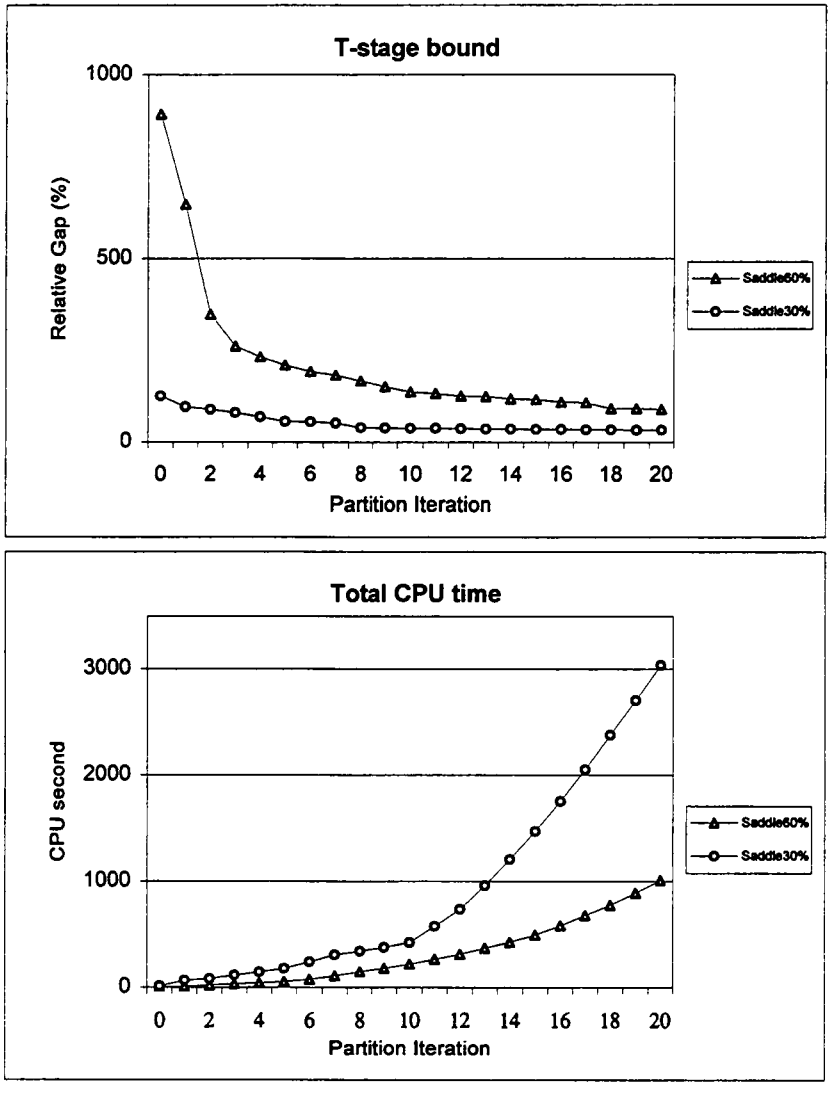


Figure 5.17: Effect of matrix density on bound convergent behavior.

5.8 Discussion and conclusion

In this research, a new solution methodology based on second-order approximations was developed for multistage stochastic programming model which has fixed technology matrices. Decomposition-based algorithms were designed to evaluate the multistage bounds more efficiently. A solution procedure using a heuristic refinement scheme partitioning under the simplicial scenario joint domain was implemented to solve multistage stochastic programming models.

The validity of the proposed solution method was established by the computational experience using several categories of randomly generated test problems. The derived multistage bounds provide valid upper and lower bounds for the multistage model and they improve as the scenario joint domains are partitioned further. After about 20 partitions, LB_T reaches 85%-90% of the optimal values on average.

The main drawback of the current solution procedure is that it takes too long to solve a problem. This solution time is strongly affected by the chosen partitioning strategy, convergence parameters, termination criteria, etc. Also, since the test bench included only the randomly generated problems with no special structure, such problems tend to be more difficult than those encountered in practice.

Furthermore, the current partitioning strategies seem to be efficient only up to 10 partitions on the scenario joint domain. After that, the improvement of the multistage bounds diminishes. As mentioned before, the diminishing effect of the bound improvement could be due to no refinement in the dynamic structure in decisions. More efficient partitioning strategies may be developed by using the notion of expected values of perfect information

(EVPI). Based on EVPI at scenario outcome nodes, the scenario sub-trees with large EVPI are recommended to be partitioned further in order to refine the dynamic structure of the scenario sub-trees. For details on EVPI, refer to Dempster [16]. The main obstacle to incorporating EVPI in a refinement scheme is that it will also require substantial amount of time in evaluating EVPI at each scenario outcome node.

To reduce the time required for evaluating the multistage bounds, multistage upper bounding problems need not be solved to optimality, as a feasible solution provides an (estimated) upper bound with which partitioning can continue. For this slight inaccuracy in the upper bound, the CPU time for solution can be improved significantly. This accuracy can be controlled via the termination criteria, in particular within the PHA algorithm for $\text{UBLP}_T(\mathcal{P})$. As an alternative, one can use a norm termination criterion which is *relative* to the norm of x_L solution instead of using a *absolute* norm termination criterion as in (5.34). The trade-off between the quality of the upper bound obtained by PHA algorithm and the time required to evaluate the multistage upper bound, then needs to be investigated.

Bibliography

Bibliography

[1] H. Attouch and R.J. Birge, Approximation and Convergence in Nonlinear Optimization, in: *Nonlinear Programming* 4, eds. O. Mangasarian, R. Meyer and S.M. Robinson (Academic Press, 1981) 367-394.

[2] M. Avriel, *Nonlinear Programming: Analysis and Methods*, Prentice-Hall, Englewood Cliffs, NJ, 1976.

[3] E.M. Beale, On Minimizing a Convex Function Subject to Linear Inequalities, Journal of the Royal Statistical Society 17B (1955) 173-184.

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

[5] J.R. Birge, Decomposition and Partitioning Methods for Multistage Stochastic Linear Programs, Operation Research 33 (1985) 989-1007.

[6] J.R. Birge, An L-shaped method computer code for multistage linear programs, in: *Numerical Techniques for Stochastic Optimization*, eds. Y. Ermoliev and R.J-B. Wets (Springer-Verlag, 1988).

[7] J.R. Birge, C.J. Donohue, D.F. Holmes, and O.G. Svintsitsli, A parallel implementation of the nested decomposition algorithm for multistage stochastic linear programs, Tech Report 94-1, Department of Industrial and Operations Engineering, University of Michigan (1994).

[8] J.R. Birge and R.J-B. Wets, Designing approximation schemes for stochastic optimization problems, in particular for stochastic programs with recourse, Mathematical Programming Study 27 (1986) 54-102.

[9] J.R. Birge and R.J-B. Wets, Stochastic Programming, Part I, II, Proceeding of the 5th International Conference on Stochastic Programming, Ann Arbor, Michigan, August 13-18, 1989, Annals of Operations Research 30-31 (1991).

[10] D.R. Cariño, T. Kent, D.H. Myers, C. Stacy, M. Sylvanus, A.L. Turner, K. Watanabe, and W.T. Ziemba. Russell-Yasuda Kasai model: an asset-liability model for a Japanese insurance company Using Multistage Stochastic Programming, Interfaces 24 (1994) 29-49.

[11] A. Charnes and W.W. Cooper, Chance-constrained Programming, Management Science 6 (1959) 73-79.

[12] G.B. Dantzig, Linear Programming Under Uncertainty, Management Science 1 (1955) 197-206.

[13] G.B. Dantzig and P. Wolfe, Decomposition Principle for Linear Programs, Operations Research 8 (1960) 101-111.

[14] G.B. Dantzig and P.W. Glynn. Parallel processors for planning under uncertainty, Annals of Operations Research 22 (1990) 1–21.

[15] M.A.H. Dempster, Introduction to Stochastic Programming, in M.A.H. Dempster (Ed.), *Stochastic Programming*, Academic Press, London, 3-59.

[16] M.A.H. Dempster, On Stochastic Programming II: Dynamic Problems Under Risk, Research Report DAL TR 86-5, Dalhousie University.

[17] J.H. Dulá, An upper bound on the expectation of simplicial functions of multivariate random variables, Mathematical Programming 55 (1992) 69–80.

[18] J.H. Dulá and R.V. Murthy, A Tchebysheff-type bound on the expectation of sublinear of polyhedral functions, Operations Research 40 (1992) 914-922.

[19] N.C.P. Edirisinghe, New second-order bounds on the expectation of saddle functions with applications to stochastic linear programming, Operations Research 44 (1996), No. 6, 909-922.

[20] N.C.P. Edirisinghe and O. Fujiwara, Optimal capacity for a multipurpose water reservoir: a chance constrained goal programming approach, Working Paper No. 89-MSC-028, Faculty of Commerce, University of British Columbia (1989).

[21] N.C.P. Edirisinghe and G-M. You, Second-order approximation and refinement in optimization under Uncertainty, Annals of Operations Research 64 (1996) 143-178.

[22] N.C.P. Edirisinghe and W.T. Ziemba, Tight bounds for stochastic convex programs, Operations Research 40 (1992) 660-677.

[23] N.C.P. Edirisinghe and W.T. Ziemba, Bounds for two-stage stochastic programs with fixed recourse, Mathematics of Operations Research 19 (1994) 292-313.

[24] N.C.P. Edirisinghe and W.T. Ziemba, Bounding the expectation of a saddle function, with application to stochastic programming, Mathematics of Operations Research 19 (1994) 314-340.

[25] N.C.P. Edirisinghe and W.T. Ziemba, Implementing bounds-based approximations in convex-concave two-stage stochastic programming, Mathematical Programming 75 (1996) 295-325.

[26] Y. Ermoliev, Stochastic quasigradient methods and their application to systems optimization, Stochastics 9 (1983) 1–36.

[27] Y. Ermoliev, Stochastic quasigradient methods and their implementation, in: *Numerical techniques for stochastic optimization*, eds. Y. Ermoliev and R.J-B. Wets (Springer-Verlag, 1988).

[28] Y. Ermoliev and R.J-B. Wets, *Numerical Techniques for Stochastic Optimization* (Springer-Verlag, 1988).

[29] Y-A. Fan and D.R. Cariño, Nested versus Nonnested Decomposition for Asset Allocation Problems, presented at the 15th International Symposium on Mathematical Programming, 1994, Ann Arbor, Michigan.

[30] K. Frauendorfer, SLP recourse problems with arbitrary multivariate distributions - the dependent case, Mathematics of Operations Research 13 (1988) 377–394.

[31] K. Frauendorfer, SLP problems: objective and right-hand side being stochastically dependent - Part II, Manuscript, IOR, University of Zurich (1988).

[32] K. Frauendorfer, A solution method for nonlinear convex stochastic optimization problems, Operational Research (1990) 761–769.

[33] K. Frauendorfer, *Stochastic Two-Stage Programming* (Springer-Verlag, Berlin, 1992).

[34] K. Frauendorfer, Barycentric scenario trees in convex multistage stochastic programming, Mathematical Programming, 75 (2, Ser. B) (1996), 277-293.

[35] K. Frauendorfer, Multistage stochastic programming: error analysis for the convex case, Zeitschrift for Operation Research, 39(1) (1994), 93-122. (1994).

[36] K. Frauendorfer and P. Kall, A solution method for SLP problems with arbitrary multivariate distributions - The independent case, Problems of Control and Information Theory 17 (1988) 177-205.

[37] H.I. Gassmann, MSLiP: A computer code for the multistage stochastic linear programming problem, *Mathematical Programming*, 47 (1990) 407-423.

[38] H. Gassmann and W.T. Ziemba, A tight upper bound for the expectation of a convex function of a multivariate random variable, Mathematical Programming Study 27 (1986) 39-53.

[39] J.L. Higle and S. Sen, Stochastic decomposition: an algorithm for two-stage stochastic linear programs with recourse, Mathematics of Operations research 16 (1991) 650-669.

[40] J.L. Higle and S. Sen, Statistical Verification of Optimality Conditions for Stochastic Programs with Recourse, Annals of Operations Research 30 (1991) 215-240.

[41] R. Horst and H. Tuy, *Global Optimization (Deterministic Approach)* (Springer-Verlag, 1990).

[42] G. Infanger, Monte Carlo (Importance) Sampling within a Benders decomposition algorithm for stochastic linear programs, Annals of Operations Research 39(1992), 69-95.

[43] J.L. Jensen, Sur les fonction convexes et les inégalités entre les valeurs moyennes, Acta Mathematica 30 (1906) 175-193.

[44] P. Kall, *Stochastic Linear Programming* (Springer-Verlag, Berlin, 1976).

[45] P. Kall, Stochastic Programming, European Journal of Operational Research 10 (1982) 125-130.

[46] P. Kall, Stochastic programming with recourse: upper bounds and moment problems - a review, in *Advances in Mathematical Optimization and Related Topics*, eds. J Guddat, P.Kall, K. Lommatzsch, M. Vlach, and K. Zimmermann, Akademia-Verlag, Berlin, 1987.

[47] P. Kall, An upper bound for SLP using first and total second moments, Annals of Operations Research 30 (1991) 267-276.

[48] P. Kall and E. Keller, GENSLP: A program for generating input for stochastic linear programs with complete fixed recourse, Manuscript, Institüt for Operations Research der Universität Zürich, Zürich CH-8044, Switzerland (1985).

[49] P. Kall, E. Keller and J. Mayer, GENSLPN: A program for generating input for stochastic linear programs, Manuscript, Institüt for Operations Research der Universität Zürich, Zürich CH-8044, Switzerland (1993).

[50] P. Kall, A. Ruszczyński, and K. Frauendorfer, Approximation techniques in stochastic programming, in: *Numerical techniques for stochastic optimization*, eds. Y. Ermoliev and R.J-B. Wets (Springer-Verlag, 1988).

[51] J. Kemperman, The General Moment Problem. A Geometric Approach. Annals of Mathematical Statistics 39 (1968) 93-112.

[52] F.V. Louveaux, A solution method for multistage stochastic programs with recourse with application to an engergy Investment problem, Operations research 28 (1980) 889-902.

[53] I.J. Lustiz, J.M. Mulvey, and T.J. Carpenter, Formulating two-stage stochastic programs for interior point methods, Operations Research, 39 (1991) 757-770.

[54] A. Madansky, Bounds on the expectation of a convex function of a multivariate random variable, Annals of Mathematical Statistics 30 (1959) 743-746.

[55] A. Madansky, Inequalities for stochastic linear programming problems, Management Science 6 (1960) 197-204.

[56] J.M. Mulvey and H. Vladimirou, Solving multistage stochastic networks: an application of scenario aggregation, Networks 21 (1991) 619-643.

[57] J.M. Mulvey and H. Vladimirou, Applying the progressive hedging algorithm to stochastic generalized networks, in: *Stochastic Programming*, Part II, eds. J.R. Birge and R.J-B. Wets, Annals Operations Research 31 (1991) 399-424.

[58] J.M. Mulvey and H. Vladimirou, Stochastic network programming for financial planning problems, Management Science 38 (1992) 1642-1664.

[59] P. Olsen, Multistage stochastic programming with recourse: the equivalent deterministic problem, SIAM Journal on Control and Optimization 14 (1976) 495-517.

[60] P. Olsen, When is a multistage stochastic programming problem well defined? , SIAM Journal on Control and Optimization 14 (1976) 518-527.

[61] P. Olsen, Multistage stochastic programming with recourse as mathematical programming in an L^p-space, SIAM Journal on Control and Optimization 14 (1976) 495-517.

[62] A. Prèkopa, Numerical solution of probabilistic constrained programming problems, in: *Numerical techniques for stochastic optimization*, eds. Y. Ermoliev and R.J-B. Wets (Springer-Verlag, 1988).

[63] S.M. Robinson, Extended scenario analysis, in: *Stochastic Programming*, Part II, eds. J.R. Birge and R.J-B. Wets, Annals Operations Research 31 (1991) 385-398.

[64] S.M. Robinson and R.J-B. Wets, Stability in two-stage stochastic programming, SIAM Journal of Control and Optimization 25 (1987)

[65] R.T. Rockafellar, Integral Functionals, Normal Integrand and Measurable Selections. In *Nonlinear Operators and the Calculus of Variations* (L. Waelbrock, Ed.), Lecture Notes in Math., No. 543 (Springer-Verlag, 1976) 157-207.

[66] R.T. Rockafellar and R.J-B. Wets, Stochastic Convex Programming: Basic Duality, Pacific Journal of Mathematics 62 (1976) 173-195.

[67] R.T. Rockafellar and R.J-B. Wets, Nonanticipativity and L^1Martingales in Stochastic Optimization Problems, Mathematical Programming study 6 (1976) 170-187.

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

[69] G. Tintner, Stochastic Linear Programming with Application to Agricultural Economics, in H.A. Antosiewicz (Ed.), *Proceedings of the Second Symposium in Linear Programming*, Washington, 1955, 197-207.

[70] R. Van Slyke and R.J-B. Wets, L-shaped linear programs with application to optimal control and stochastic optimization, SIAM Journal on Applied Mathematics 17 (1969) 638–663.

[71] R. J-B. Wets, Stochastic programs with fixed recourse: the equivalent deterministic program, SIAM Review 16 (1974) 309-339.

[72] R.J-B. Wets, Convergence of convex functions, variational inequalities and convex optimization problems, in: *Variational Inequalities and Complimentary Problems*, eds. R.W. Cottle, F. Giannesi, and J-L. Lious (J. Wiley and Sons, New York, 1980) 375–403.

[73] R.J-B. Wets, Solving Stochastic Programs with Simple Recourse, Stochastics 10 (1983) 219-242.

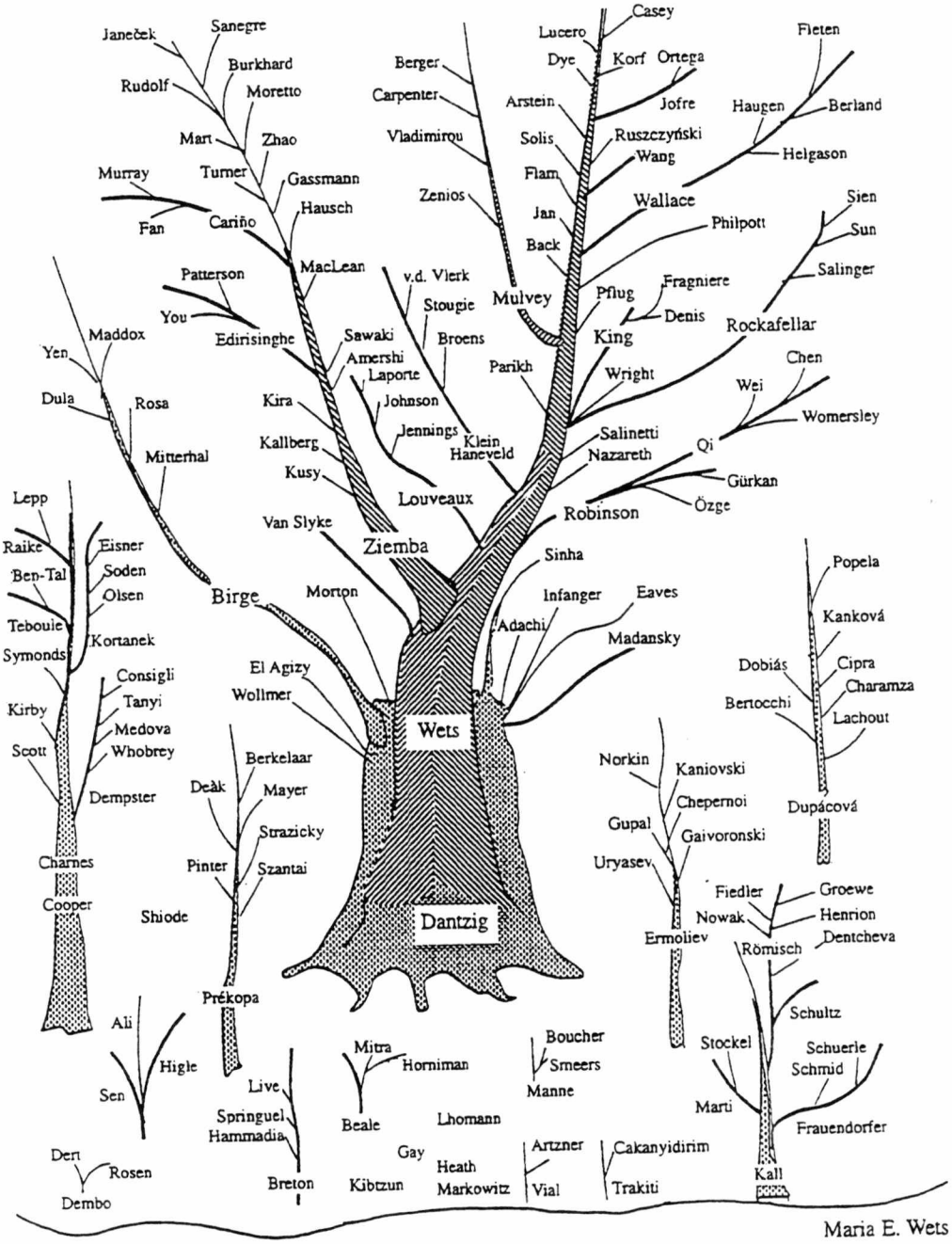
[74] R.J-B. Wets, Stochastic programming: solution techniques and approximation schemes, in: *Mathematical Programming: State of the art*, eds. A. Bachem, M. Grõtschel and B. Korte (Springer-Berlag, Berlin, 1983) 566-603.

[75] R.J-B. Wets, Large Scale Linear Programming Techniques, in: *Numerical techniques for stochastic optimization*, eds. Y. Ermoliev and R.J-B. Wets (Springer-Verlag, 1988) 63-93.

[76] W.T. Ziemba, Stochastic Programs with Simple Recourse, in: *Mathematical Programming: Theory and Practice*, eds. P.L. Hammer and G. Zoutendijk (North-Holland, Amsterdam, 1974) 213-273.

Appendices

Appendix A: Stochastic programming's orchard 1998



Source: Annals of Operations Research, Volume 85, 1999

Appendix B: Preliminary definitions of terms and concepts

This appendix provides some preliminary definitions of terms and concepts that will be used throughout this thesis.

convex set A subset $K \subset \Re^n$ is said to be a convex set if every convex combination of any pair of points in K is also in K. That is, if $\tilde{x} \in K$ and $\hat{x} \in K$, then $\alpha\tilde{x} + (1-\alpha)\hat{x} \in K$ for all $0 \leq \alpha \leq 1$.

convex function Let $f(x)$ be a real-valued function defined over points $x = (x_1, \ldots, x_n)^T \in \Gamma \subset \Re^n$, where Γ is either $\Re^n$ or a convex subset of $\Re^n$. The $f(x)$ is said to be a convex function iff for any $x^1 = (x_1^1, \ldots, x_n^1)$, $x^2 = (x_1^2, \ldots, x_n^2)^T \in \Gamma$ and $0 \leq \alpha \leq 1$, we have

$$f(\alpha x^1 + (1-\alpha)x^2) \leq \alpha f(x^1) + (1-\alpha)f(x^2)$$

Similarly, if $g(x)$ is a real-valued function defined on the convex subset Γ of $\Re^n$, it is said to be a concave function iff for any $x^1, x^2 \in \Gamma$, and $0 \leq \alpha \leq 1$, we have

$$g(\alpha x^1 + (1-\alpha)x^2) \geq \alpha g(x^1) + (1-\alpha)g(x^2)$$

affine function A real-valued function $\theta(x)$ defined on a convex subset $\Gamma \subset \Re^n$ is said to be an affine function iff it is both convex and concave; that is, if it satisfies

$$\theta(\alpha x^1 + (1-\alpha)x^2) = \alpha\theta(x^1) + (1-\alpha)\theta(x^2)$$

for any $x^1, x^2 \in \Gamma$, and $0 \leq \alpha \leq 1$. Given the affine function $\theta(x)$, it can be shown that there exist real numbers $c_0, c_1, \ldots, c_n$ such that $\theta(x) = c_0 + c_1x_1 + \ldots + c_nx_n$.

proper convex function A convex function f(x) is said to be proper if it is finite valued for some x and does not go to negative infinity on the domain.

deterministic LP problem In an LP we have to optimize the linear objective function subject to linear equality and/or inequality constraints, and sign restrictions on the variables. If there are no inequality constraints and the variables are restricted to be nonnegative, the LP will be of the form:

$$\text{Minimize} \qquad z(x) = cx$$

$$\text{Subject to} \quad Ax = b,\ x \geq 0$$

where A, b, c are given matrices or vectors with constant coefficients. The **dual problem** of the above LP is

$$\text{Maximize} \qquad v(x) = \pi b$$

$$\text{Subject to} \quad \pi A \leq c,\ \pi \text{ unrestricted}$$

where π is the row vector of dual variables. Each dual variable is associated with each constraint in the primal. The dual variables are also called simplex multipliers or Lagrange multipliers. In the above primal-dual pair of LPs, the **weak duality theorem** states that $z(x) \geq v(\pi)$, where x is a feasible solution of the above primal LP and π is a feasible solution of the above dual solution. The **strong duality** theorem states that if either the primal or the dual problem has an optimal feasible solution, then the other does also, and the two optimal objective values are equal.

quadratic program General quadratic programming problems have a quadratic objective function and linear or quadratic constraints. Quadratic programming is typically used in investment portfolio analysis. OSL addresses quadratic programming (QP) problems where the objective function is quadratic and convex and the constraints are linear. The primal

problem is as stated below.

minimize c'x + 0.5x'Qx

subject to: $Ax = b$ $\quad x \geq 0$

Here c, x, A, and b are as defined above. Q is a positive semi-definite matrix of dimension n * n. Quadratic programming problems can be solved using either simplex-based or interior-point methods. The QP solver in OSL uses a simplex-based approach.

Vita

Guey-Mei You was born in Taipei, Taiwan, the Republic of China. She graduated from Taipei Municipal First Girl Senior High School in June 1979. She received her Bachelor of Science in Psychology in June 1983. After working as a research assistant for two years in the Department of Psychology at National ChengChi University and the Department of Fine Arts at National Institute of the Arts, she went to Purdue University, West Lafayette, Indiana, USA, to pursue postgraduate degrees. She received her Master of Science in Quantitative-Mathematical Psychology in August 1989, and received her Master of Science in Applied Statistics in May 1990. She then joined her husband in Knoxville, Tennessee to pursue her doctoral degree. She received the doctoral degree in Management Science in May 1999.