5-2001

Decomposition techniques for large scale stochastic linear programs

Earl Ike Patterson

To the Graduate Council:

I am submitting herewith a dissertation written by Earl Ike Patterson entitled "Decomposition techniques for large scale stochastic linear programs." I have examined the final electronic 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.

Chanaka Edirisinghe, Major Professor

We have read this dissertation and recommend its acceptance:

Accepted for the Council:

Carolyn R. Hodges

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

To the Graduate Council:

I am submitting herewith a dissertation written by Earl I. Patterson entitled "Decomposition Techniques for Large Scale Stochastic Linear Programs". 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.

Chanaka Edirisinghe, Major Professor

We have read this dissertation
and recommend its acceptance:

Accepted for the Council:

Interim Vice Provost and
Dean of the Graduate School

Decomposition Techniques for Large Scale Stochastic Linear Programs

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

Earl I. Patterson
May 2001

DEDICATION

This dissertation is dedicated to my parents

Earl H. (deceased) and Euretha M. Patterson

for a lifetime of love, devotion, and support!

ACKNOWLEDGMENTS

My gratitude is owed to a great many people who have made my time at the University of Tennessee so personally rewarding. I am deeply indebted to Dr. Chanaka Edirisinghe for his contributions to the work culminating in this dissertation. Dr. Edirisinghe's guidance and encouragement have been invaluable to me in all phases of my graduate studies at the University of Tennessee. I have come to value his opinions on both a professional and personal level.

Special recognition is also owed to the other members of my Dissertation Committee, Dr. Melissa Bowers, Dr. Hamparsum Bozdogan, and Dr. Charles Noon. I am extremely grateful for the time and effort each expended in reviewing and suggesting improvements to this manuscript. I also want to thank Dr. Bozdogan for his assistance in my research into Mahalanobis squared distances.

ABSTRACT

Stochastic linear programming is an effective and often used technique for incorporating uncertainties about future events into decision making processes. Stochastic linear programs tend to be significantly larger than other types of linear programs and generally require sophisticated decomposition solution procedures. Detailed algorithms based upon Dantzig-Wolfe and L-Shaped decomposition are developed and implemented. These algorithms allow for solutions to within an arbitrary tolerance on the gap between the lower and upper bounds on a problem's objective function value. Special procedures and implementation strategies are presented that enable many multi-period stochastic linear programs to be solved with two-stage, instead of nested, decomposition techniques. Consequently, a broad class of large scale problems, with tens of millions of constraints and variables, can be solved on a personal computer. Myopic decomposition algorithms based upon a short-sighted view of the future are also developed. Although unable to guarantee an arbitrary solution tolerance, myopic decomposition algorithms may yield very good solutions in a fraction of the time required by Dantzig-Wolfe/L-Shaped decomposition based algorithms. In addition, derivations are given for statistics, based upon Mahalanobis squared distances, that can be used to provide measures for a random sample's effectiveness in approximating a parent distribution. Results and analyses are provided for the applications of the decomposition procedures and sample effectiveness measures to a multi-period market investment model.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1
Introduction

Managers at most levels of business, government, and industry must make present-day decisions with imperfect knowledge of future events. Consider, for example, a property insurance provider that returns a maturity refund to the client in addition to the protection against damage and/or loss normally provided. Insurance policies must be structured and assets/liabilities allocated based upon uncertainties in how the future unfolds. Decisions must be made *here-and-now* before uncertainties in future interest rates, the economy, and liabilities are resolved. Sufficient information may be, however, available to make probabilistic statements concerning future uncertainties. Such information might include historical data and expert judgment. The insurance company, therefore, desires to develop and implement an asset/liability management model that would allow the company to make asset allocation decisions that hedge against uncertainties in future events.

One method for guiding here-and-now decisions while considering the future uncertainty explicitly is the *stochastic programming* field of mathematical programming. Models based upon stochastic programming generally prescribe plans that involve a trade-off between costs associated with long-term anticipatory or decisions and the costs associated with short-term recourse or adaptive decisions (see Edirisinghe [62]). The above insurance company example is a real-world case study and it is described in further detail below (Section 1.2.1).

Chapter 1 Introduction

Relatively few applications using stochastic programming techniques have been reported in published material until recently although this discipline has existed since 1955. A major drawback to stochastic programming modeling is that the resulting problem size for many practical applications is very large. Consequently, solution complexity of such models often hinders implementations of stochastic programming decision models. The focus of this thesis is the development and implementation of solution procedures for stochastic programming models.

The remainder of this chapter is organized into five sections. The first two sections summarize the historical background of stochastic programming and provide example applications from the literature. Papers cited in those two sections are from a representative, but small, subset of available documentation. A far more comprehensive stochastic programming bibliography (over 3700 entries) is given by M. van der Vlerk [199]. Thesis scope and general notational conventions are described in the third and fourth sections. Organization of the thesis is the topic of the fifth and final section.

1.1 Historical Background

This section summarizes the evolution of stochastic programming theory and solution procedures over the last five decades of the twentieth century. Three periods are used to document the historical information – early history covering the fifties through the seventies, developments in the eighties, and the recent history of the nineties.

1.1.1 Early History

The birth of stochastic programming as a field of mathematical programming followed shortly after the introduction of linear programming. According to Dantzig [44, page 4], the articles published in 1949 by Wood and Dantzig [209] and Dantzig [40] were the first two formal papers about linear programming (introduced in 1947 – see Dantzig [44, page 4] and Dantzig and Thapa [48, page xxii]). Techniques for incorporating uncertainty into linear programs were introduced independently six years later by Beale [8] and Dantzig [41]. The procedures and problem formulations presented by Beale and Dantzig have since become known as *stochastic programming with recourse* and form one of the two major subfields of stochastic programming. Charnes, Cooper, and Symonds [30] in 1958 and Charnes and Cooper [29] in 1960 introduced the second subfield which they named *chance-constrained programming*. The remainder of this thesis is concerned only with stochastic programming with recourse and the interested reader can consult Kall [117, Chapter 4], Kall and Wallace [118, Chapter 4], and Prekopa [169, Chapter 8] for more information on chance-constrained programming.

The first procedure for solving large-scale linear programs in a reasonable amount of time uses the *decomposition principle* presented by Dantzig and Wolfe, [49] and [50], in 1960 and 1961. Inspiration for the decomposition principle was provided by the work of Ford and Fulkerson [79] in 1958 on multi-commodity network flow problems (see Dantzig and Wolfe [49, page 102]). Procedures based upon the decomposition principal have since become known as Dantzig-Wolfe decomposition (DWD). Dantzig and Madansky [47] in

1961 and Madansky [138] in 1963 were the first to use DWD specifically for a stochastic programming problem. A second decomposition procedure, generally referred to as the L-Shaped decomposition (LSD) method, was introduced in 1969 by Van Slyke and Wets [196] based upon work by Kelly [120] (1960) on cutting planes and Benders decomposition [9] (1962) of mixed integer programs. Van Slyke and Wets also contributed to stochastic programming theory with four papers published in 1966 – Van Slyke and Wets [195] and Wets [202], [203], and [204].

Papers published in the seventies served primarily to consolidate and expand upon the theory and solution procedures developed during the previous two decades. Geoffrion, [87] and [88], gave a synthesis in 1970 of existing theory and algorithms. Solution algorithms for convex stochastic programs with recourse were presented in 1970 by Ziemba [215]. Eaves and Zangwill [61] (1971), Geoffrion [89] (1972), and Hogan [104] (1973) provided additional pioneering work on cutting plane theory and Benders decomposition. Algorithms for applying DWD to a problem involving multiple (more than two) periods or stages were given in 1973 by Glassey [90] and in 1974 by Ho and Manne [102]. These algorithms are usually referred to as *nested decomposition* algorithms (since the procedures for one period or group of periods is nested within the procedures for a previous period or group of periods) and are based upon concepts first introduced by Dantzig and Wolfe in their original decomposition paper [49, pages 109 - 110]. Dupacova [57] (1974) provided a theoretical basis for stochastic programs with non-convex, non-separable penalty functions. Additional theory for problems spanning multiple stages was presented in 1974

by Wets [205] and in 1976 by Olsen, [160] and [161]. Kall [117] wrote one of the first books, published in 1976, devoted to stochastic linear programming. Huang, Ziemba, and Ben-Tal [106] provided refinements in 1977 to bounds-based approximations to stochastic programs based upon the classic bounds of Jensen and Edmondson-Madansky (see Birge [17, page 288] and Edirisinghe [62, pages 21 and 25]).

1.1.2 Eighties

The eighties ushered in a two-decade period of explosive growth in stochastic programming theory, algorithms, and applications. This growth, not surprisingly, parallels the advances made in the personal computer and distributed computing industries.

Advances in nested decomposition based upon DWD were made in the early eighties by Abrahamson [1], Ament, et al. [3], Birge [12], and Ho and Loute [100] and [101]. Nested decomposition based upon LSD was introduced in 1980 by Louveaux [133] and expanded upon in the last half of the decade by Birge [11], Birge and Louveaux [16], Gassmann [82], [83], and [84], Louveaux [134], and Wittrock [207].

Bounds-based approximation schemes were enhanced by Wets [206] (1983) and Birge and Wets [19] (1986) and [20] (1987). Dupacova [58] (1987) presented methods for analyzing stochastic programs when there is incomplete knowledge of the underlying probability distribution. Ruszczynski [179] introduced *regularized decomposition* in 1986 as a method to possibly improve decomposition efficiency by adding a quadratic regularizing term to the objective function of a stochastic problem. Glynn and Iglehart [92] (1989)

proposed methods for incorporating the classical variance reduction technique of importance sampling into stochastic programming problems.

Birge, et al. [13] made a significant contribution in 1987 with a proposal for a standard input format for stochastic programs. The proposed standard provides extensions to the *mathematical programming system* (MPS) input file system for linear programs (see reference manuals [107, Chapter 9] and [109, Appendix E] for details on the MPS standard).

1.1.3 Nineties

Many advances in existing solution procedures for stochastic programs were made in this decade. Ruszczynski [182] and [183] and Ruszczynski and Swietanowski [184] expanded on Ruszcznski's regularized decomposition method. Infanger and Morton [113] proposed procedures for sharing cuts in the L-Shaped decomposition of stochastic linear programs with interstage dependency. Enhancements were made to bounds-based approximations by Edirisinghe [62], [63], and [65], Edirisinghe, Atkins, and Iyogun [67], Edirisinghe and You [68], Edirisinghe and Ziemba [69], [70], [71], and [72], Morton and Wood [149], and Rosa and Takriti [177]. Rosa and Ruszczynski [176] and Ruszczynski [181] proposed improved solution procedures based upon augmented Lagrangian decomposition. Additional importance sampling techniques were suggested by Dantzig and Glynn [45], Dempster and Thompson [53], Infanger [111] and [112], and Morton [148]. Gassman and Ireland [85] considered extensions to algebraic modelling languages for stochastic linear

programs. Gassmann and Schweitzer [86] proposed improvements to the standard input format for stochastic programs.

Several additional concepts were also introduced and/or developed during the nineties. Application of parallel computing techniques to stochastic programs was covered by Birge, et al. [14], Dantzig and Glynn [45], Dempster and Thompson [53], Electric Power Research Institute Report EL-6769 [75], Korycki [129], Mulvey and Ruszczynski [152] and [153], Nielsen and Zenios [158], Ruszczynski [180], Vladimirou [197], and Vladimirou and Zenios [198]. Interior point solution procedures were documented by Bahn, et al. [5], Birge and Holmes [15], Kim and Nazareth [122], Lustig, Mulvey, and Carpenter [137], Messina and Mitra [144], Meszaros [145], and Zakeri, Philpott, and Ryan [213]. Higle and Sen [97] and [98] introduced the *stochastic decomposition* algorithm for solving stochastic programs. Extensions to the stochastic decomposition algorithm were subsequently given by Chen and Powell [31], Higle, Lowe, and Odio [96], Morton [148], and Yakowitz [211] and [212]. Rockafellar and Wets [174] introduced the *progressive hedging* algorithm solution procedure based upon scenario and policy aggregation. Subsequent techniques using scenario aggregation were proposed by Chun and Robinson [32] and Kiwiel, Rosa, and Ruszczynski [125].

1.2 Applications

Stochastic programming techniques have been applied to the decision making processes of a broad range of organizations in business, government, and industry. Such applications have increased significantly in the past two decades due in large measure to advances in computing technology. This section will summarize a representative sample of stochastic programming application articles published within the past twenty years. Four application categories are used: investment planning, electric power generation, process control, and production management. Note that these categories are not intended to be definitive classification groups, but are defined for descriptive purposes only. Most articles, in fact, could be placed in two or more of the categories and several could be placed in all four categories. One paper in each category will be summarized in some detail while the remaining articles in that group will be listed in a table.

1.2.1 Investment Planning

The introductory example is actually a synopsis of a multiple period stochastic linear programming model developed by the Frank Russell Company for the Yasuda Fire and Marine Insurance Company, Ltd. (see Carino, et al. [26], Carino, Myers, and Ziemba [27], and Carino and Ziemba [28]). The Russell-Yasuda Kasai (*kasai* means fire in Japanese) model was developed to replace an existing static mean-variance asset allocation model. Discrete probability distributions are used in the Russell-Yasuda Kasai (RY) model to account for uncertainties in returns, interest rates, liabilities, and other stochastic variables. Extra in-

come of 79 million dollars (US) was realized during the first two years (fiscal 1991 and 1992) that the RY model was employed.

Summary descriptions of the RY model and other investment planning application papers are provided in Table 1.1.

1.2.2 Electric Power Generation

Jacobs, et al. [114] describe a multiple period stochastic linear programming model developed at Pacific Gas and Electric Company (PG&E) to optimize monthly hydrogeneration scheduling over a 24 month planning horizon. Hydrogeneration at PG&E was scheduled with a deterministic network optimization model prior to development of the stochastic programming model. Discrete probability distributions are used to model uncertainties, especially the uncertainties in streamflows, in the stochastic model. The authors report that expert users were satisfied with the results of the stochastic programming model during the initial testing and implementation phases (1992 - 1994).

Table 1.2 contains short descriptions of the above and other papers from this application category.

1.2.3 Process Control

Paules and Floudas [162] develop a stochastic mixed integer programming model to optimize the synthesis strategy of heat integrated distillation sequences with a single multicomponent feed stream. The stochastic model uses discrete probability distributions to account

Table 1.1: Investment Planning Applications

Authors	Cite	Year	Description
Anandalingam	[4]	1987	Investment decisions in iron/steel industry in India
Birge, Rosa	[18]	1995	Multi-period model of investment uncertainty in the costs of global CO_2 emission policy
Carino, et al.[a]	[26]	1994	Japanese insurance company multi-period asset/liability model
Carino, et al.[b]	[27]	1998	Formulation details for the above model
Carino, Ziemba	[28]	1998	Concepts and technical issues for the above model
Consigli, Dempster	[38]	1998	Portfolio management for a pension fund
Dantzig, Infanger	[46]	1993	Multi-period portfolio management
Dupacova, et al.[c]	[59]	1997	Bond portfolio management model
Frauendorfer	[81]	1996	Optimal funding by borrowing bonds of different maturities
Golub, et al.[d]	[94]	1995	Two-period portfolio management
Henaff	[95]	1998	Two-period exotic derivatives investments
Hiller, Eckstein	[99]	1993	Asset/liability management with interest rate contingent claims
Kira, Kusy	[124]	1990	Optimal project selection for capital expenditures
Klaassen	[126]	1998	Synthesis of asset/liability management models/pricing theory
Kusy, Ziemba	[131]	1986	Multi-period bank asset/liability management model
Mulvey, et al.[e]	[151]	1997	Procedures and example cases for financial risk management
Mulvey, Vladimirou	[154]	1992	Financial planning using networks with stochastic parameters
Pieptea	[166]	1987	Optimize holdings in bonds/funds with stochastic interest rates
Prisman, et al.[f]	[170]	1986	Two-period bank asset/liability management model
Wagner, et al.[g]	[200]	1994	Two-period model for optimal placement and operation of pumping wells to contain groundwater contamination
Watanabe, Ellis	[201]	1993	Two-period model to minimize costs of acid rain control
Zenios, et al.[h]	[214]	1998	Fixed-income portfolio management with uncertain interest rates

[a] Carino, Kent, Myers, Stacy, Sylvanus, Turner, Watanabe, Ziemba

[b] Carino, Myers, Ziemba

[c] Dupacova, Bertocchi, Moriggia

[d] Golub, Holmer, McKendall, Pohlman, Zenios

[e] Mulvey, Rosenbaum, Shetty

[f] Prisman, Slovin, Sushka

[g] Wagner, Shamir, Marks

[h] Zenios, Holmer, McKendall, Vassiadou-Zeniou

Table 1.2: Electrical Power Generation Applications

Authors	Cite	Year	Description
Bloom	[22]	1983	Least-cost generation capacity expansion model
Bloom, et al.[a]	[23]	1984	Implementation and experience with above model
Borison, et al.[b]	[25]	1984	Least-cost generation capacity expansion model
EPRI[c]	[74]	1989	Resource expansion plan for large-scale multi-area electrical power generation and transmission system
Hobbs, Ji	[103]	1999	Minimize operation costs for a multi-area electrical power generation and transmission system
Jaco, et al.[d]	[114]	1995	Optimal scheduling of hydrogeneration for a large utility company
Morton	[147]	1996	Multi-period hydroelectric scheduling
Pereira, Pinto	[164]	1985	Minimize expected operation costs for a Brazilian multiple reservoir hydrothermal system
Pereira, Pinto	[165]	1991	Expansion of the above model with application case study
Qiu, Girgis	[172]	1993	Maximize reliability of electric power generation system
Romi, Schultz	[175]	1996	Optimize electrical power generation in a system of thermal power and pumped storage plants
Rotting, Gjelsvik	[178]	1992	Optimal seasonal scheduling for power generation in the Norwegian power system
Sanghvi, Shavel	[185]	1986	Hydroelectric generation capacity expansion with uncertain hydro energy availability and uncertain load growth
Silva, et al.[e]	[189]	1995	Minimum cost maintenance schedule for generating units in a multi-area hydroelectric system
Takriti, et al.[f]	[192]	2000	Minimize electricity generation costs of electric utilities with uncertain load demand and uncertain spot prices

[a] Bloom, Caramanis, Charny

[b] Borison, Morris, Oren

[c] Electric Power Research Institute, Palo Alto, CA

[d] Jacobs, Freeman, Grygier, Morton, Schultz, Staschus, Stedinger

[e] Silva, Morozowski, Fonseca, Oliveira, Melo, Mello

[f] Takriti, Krasenbrink, Wu

for uncertainty in feed stream flowrate and component composition changes over a finite number of periods of operation within a chemical plant. Paules and Floudas developed the stochastic model to improve upon synthesis strategies determined with a single period model with fixed flowrate and component composition.

Short descriptions of the above and other process control articles are provided in Table 1.3.

1.2.4 Production Management

Eppen, Martin, and Schrage [73] describe a stochastic mixed integer programming capacity planning model developed for General Motors (GM) to use in making decisions concerning four of their automobile lines. The model has a planning horizon of five one-year periods with three possible outcomes representing uncertain demand during each period. This approach agreed with traditional GM forecasting of demand as either pessimistic, standard, or optimistic. The authors indicate that results of the stochastic programming model were instrumental in motivating GM to conduct further cost and forecasting analyses of the four automobile lines.

Table 1.4 provides summary descriptions of the above GM model paper as well as other papers in the production management application category.

Table 1.3: Process Control Applications

Authors	Cite	Year	Description
Alonso, et al.[a]	[2]	2000	Optimize air traffic flow management
Dror	[55]	1993	Multi-period model of the vehicle routing problem with uncertain demands
Duffuaa, Al-Sultan	[56]	1999	Multi-period model for scheduling maintenance personnel
Dupacova, et al.[b]	[60]	1991	Multi-period model for upgrading and expanding a water resources management system in Czechoslovakia
Fernandez, et al.[c]	[76]	1998	Multi-period model for project scheduling with stochastic task durations
Glockner, Nemhauser	[91]	2000	Multi-period model for a network flow problem with stochastic arc capacities
Hsu, Bassok	[105]	1999	Multiple product inventory problem with downward substitution
Ierapetritou, et al.[d]	[108]	1996	Optimal design for process models involving stochastic parameters
Martel, et al.[e]	[143]	1990	Two-period model to determine the distribution of shell fragments
Paules, Floudas	[162]	1992	Optimize a chemical process heat integration scheme with stochastic feed composition and flowrate
Pistikopoulos, Ierapetritou	[167]	1995	Two-period model for the optimal design of a chemical processing plant
Pistikopoulos, et al.[f]	[168]	1996	Optimal design, schedule, and maintenance plan for a chemical batch plant
Qi	[171]	1985	Two-period model for a transportation problem with stochastic demands
Sapountzis	[186]	1989	Optimize the allocation of units of blood from a regional blood transfusion center to area hospitals
Wollmer	[208]	1985	Two-period model for critical path planning with uncertain job completion times

[a] Alonso, Escudero, Ortuno

[b] Dupacova, Gaivoronski, Kos, Szantai

[c] Fernandez, Armacost, Per-Edwards

[d] Ierapetritou, Acevedo, Pistikopoulos

[e] Martel, Nadeau, Price

[f] Pistikopoulos, Thomaidis, Melin, Ierapetritou

Table 1.4: Production Management Applications

Authors	Cite	Year	Description
Bienstock, Shapiro	[10]	1988	Two-period model to optimize resource acquisition policies
Bitran, Dasu	[21]	1992	Optimize ordering policies for a production process with stochastic yields and hierarchy of grades of output
Clay, Grossmann	[35]	1997	Two-period model to optimize production planning in a chemical processing plant
Couillard	[36]	1993	Trucking company decision support system incorporating the model below
Couillard, Martel	[37]	1990	Two-period model to optimize the size and composition of a vehicle fleet
Darby-Dowman, et al.[a]	[51]	2000	Two-period model to optimize planting plans for a vegetable crop
Eppen, et al.[b]	[73]	1989	Multi-period model to optimize capacity planning for four automobile models
Fine, Freund	[77]	1990	Two-period model to optimize the investment in flexible manufacturing capacity
Jonsson, Silver	[116]	1989	Two-period model to optimize the inventory of components common to several products
Jonsson, et al.[c]	[115]	1993	Extensions to the above inventory model
King, et al.[d]	[123]	1988	Two-period model for management of eutrophication of Lake Balaton in Hungary
Louveaux, Peeters	[135]	1992	Two-period model for the uncapacitated facilities location problem with uncertainty on demand, selling prices, production, and transportation
Sinha, Wei	[190]	1992	Two-period model to optimize production capacity/levels in discrete part manufacturing
Somlyody, Wets	[191]	1988	Detailed model description and analysis for the Lake Balaton eutrophication model (see King, et al. above)

[a] Darby-Dowman, Barker, Audsley, Parsons

[b] Eppen, Martin, Schrage

[c] Jonsson, Jornsten, Silver

[d] King, Rockafellar, Somlyody, Wets

1.3 Scope

Stochastic programming techniques can be applied to both linear and nonlinear problem formulations. This thesis is concerned only with stochastic linear programs with recourse. Solution procedures for these types of problems were classified by Edirisinghe [65] as belonging to one of three categories: sampling-based approximation techniques, bounds-based approximations, or mathematical decomposition of the grand linear program. Sampling-based methods such as the stochastic decomposition algorithm of Higle and Sen (see [97] and [98]) use iteratively drawn random samples from the underlying probability distribution for a decomposition algorithm or to compute stochastic quasi-gradients. Bounds-based approximation algorithms (e.g., see Birge and Wets [19] and Huang, Ziemba, and Ben-Tal [106] for general details) use a successive approximation procedure based upon computable bounds on the objective function value. The focus of this thesis is the third technique – using mathematical decomposition procedures to solve the grand linear formulation of the stochastic programming problem. Specifically, innovations to improve the efficiency of mathematical decomposition methods, particularly L-Shaped and Dantzig-Wolfe decomposition, are described herein. A decomposition algorithm incorporating a myopic view of the future and a sampling effectiveness measurement scheme based upon Mahalanobis distances are also developed.

1.4 Notation

General notational conventions that are used throughout the remainder of this thesis are defined below. Specific notational constructs are defined when introduced.

Scalars will be represented by italicized text with small letters generally used for indices and large letters for fixed quantities or limits, e.g. $j = 1, \cdots, J$. Arrays will be shown in bold upright text with small letters representing vectors and large letters representing matrices, e.g. vector $\mathbf{x}$ and matrix $\mathbf{A}$. Array transposition will be indicated by a prime symbol placed prior to any identifying superscripts, e.g. $\mathbf{x}'^{(1)} = \left(\mathbf{x}^{(1)}\right)'$. Array dimensions will generally not be explicitly stated except where deemed necessary for clarity. All arrays in any expression involving multiple arrays are assumed to be compatible. The identity matrix is represented by I which is assumed appropriately dimensioned for each use. When appropriate, the dimension of the identity matrix will be denoted by a subscript, e.g. $\mathrm{I}_M \in \mathbb{R}^{M \times M}$. Vectors of all zeros and all ones are represented by $\mathbf{0}$ and $\mathbf{1}$ respectively where dimensions are to be compatible for the expression in which they are used. A specific column or row of a matrix is denoted using the notation of Dantzig and Thapa [48] – column j of matrix $\mathbf{A}$ is denoted by $\mathbf{A}_{\bullet j}$ while row i of matrix $\mathbf{A}$ is $\mathbf{A}_{i\bullet}$.

Functions are represented by upper case letters in Fractur font, $\mathfrak{Q}(\mathbf{x})$, or Calligraphic font, $\mathcal{Q}(\mathbf{x})$. Sets are shown by upper case letters in Sans Serif font, e.g. $\mathsf{X} = \{\mathbf{x} | \mathbf{x} \geq \mathbf{0}\}$ and $\mathsf{J} = \{1, \ldots, J\}$. The phrase *for all* is indicated by the symbol $\forall$ and the operation $|\bullet|$ represents the absolute value of the operand.

1.5 Organization

The remainder of this thesis consists of six additional chapters and eight appendices. Chapters 2 and 3 develop the foundations for stochastic linear programs and decomposition of general linear programs respectively. Mathematical decomposition procedures for stochastic linear programs are then derived in Chapter 4. Formulation of a multiple period market investment model, software to solve the model, and procedures to measure sample effectiveness are described in Chapter 5. Chapter 6 contains results obtained by applying decomposition techniques to the market investment model. Conclusions and summary information are presented in the seventh and final chapter.

Material supplemental to one of the chapters is given in each of the first seven appendices. Supplemental information is provided for decision tree node mapping (supplemental to Chapter 2), myopic decomposition (Chapter 4), piece-wise linear approximations to fixed costs in the market investment model (Chapter 5), sizes of model problems (Chapter 5), solution procedures for decomposition subproblems (Chapter 5), measures of sample effectiveness (Chapter 5), and decomposition results graphs (Chapter 6). A descriptive listing of all acronyms used in this thesis is at Table H.1 in the eighth and final appendix.

Chapter 2
Stochastic Linear Programs With Recourse

Stochastic linear programs with recourse are probably the most widely applied stochastic optimization models. The notation and formulations for problems of this type are defined in the four sections of this chapter. Generally accepted notational conventions and formulations for two-period models are discussed in the first section. Decision trees and the notation used for stochastic programs over an arbitrary but finite number of periods are described in the second section. Results from the first two sections are coupled in the third section to define formulations for stochastic linear programs with recourse over multiple periods. The final section provides details on stochastic programs that exhibit a special structural characteristic referred to as block-separable recourse.

2.1 Two-Period Stochastic Programs

A two-period stochastic program is used to model a *decision-observation-decision* process. The first decision must be made prior to the occurrence of some influential event while the second decision provides recourse after observation of the event. Knowledge of the data associated with the event is encompassed by the probability space $(\Omega, \mathfrak{F}, \mathcal{P})$. The universal set Ω contains all possible values or outcomes for the uncertain event. Field $\mathfrak{F}$ is a family of measurable subsets of Ω and $\mathcal{P}$ is a probability measure defined on $\mathfrak{F}$. Greater detail on probability spaces is available in Birge and Louveaux [17, Section 2.1], Chung [33,

Appendix 1 to Chapter 4], Degroot [52, Section 2.3], Kall and Wallace [118, Section 1.3], Lindgren [132, Sections 1.4 and 2.1], and Taylor and Karlin [194, Section 1.2.8].

Stochastic programming formulations for the cases of a general probability measure and a finitely denumerable probability measure are discussed below.

2.1.1 General Probability Measure

Let $\mathcal{E}_{\xi(\omega)}(\bullet)$ denote the expectation operator of the enclosed operand with respect to the random vector $\boldsymbol{\xi}(\omega)$, $\omega \in \Omega$. Then the general formulation for a two-period stochastic program with recourse is:

$$\begin{array}{rrcrcll} z^* = \max & \mathbf{c}'^{(1)}\mathbf{x}^{(1)} & + & \mathcal{E}_{\xi(\omega)}\left[\mathbf{c}'^{(2)}(\omega)\,\mathbf{x}^{(2)}(\omega)\right] & & & \\ \text{s.t.} & \mathbf{A}\mathbf{x}^{(1)} & & & \leq & \mathbf{b}^{(1)}, & \\ & \mathbf{B}(\omega)\,\mathbf{x}^{(1)} & + & \mathbf{W}(\omega)\,\mathbf{x}^{(2)}(\omega) & \leq & \mathbf{b}^{(2)}(\omega), & \omega \in \Omega, \text{ a.s.} \quad [2.1] \\ & & & \mathbf{x}^{(1)} & \geq & \mathbf{0}, & \\ & & & \mathbf{x}^{(2)}(\omega) & \geq & \mathbf{0}, & \omega \in \Omega, \text{ a.s.} \end{array}$$

where one or more of the arrays $\mathbf{b}^{(2)}(\omega)$, $\mathbf{c}^{(2)}(\omega)$, $\mathbf{B}(\omega)$, and $\mathbf{W}(\omega)$ may be stochastic (one or more elements of these arrays may be stochastic) and the constraints involving ω hold almost surely (a.s.). Decision vector $\mathbf{x}^{(2)}(\omega)$ may assume different values for each $\omega \in \Omega$. Arrays $\mathbf{b}^{(1)}$, $\mathbf{c}^{(1)}$, and $\mathbf{A}$ are known fixed arrays (not stochastic) and $\mathbf{x}^{(1)}$, $\mathbf{c}^{(1)} \in \mathbb{R}^{N_1}$, $\mathbf{b}^{(1)} \in \mathbb{R}^{M_1}$, and $\mathbf{A} \in \mathbb{R}^{M_1 \times N_1}$. For each $\omega \in \Omega$, $\mathbf{x}^{(2)}(\omega)$, $\mathbf{c}^{(2)}(\omega) \in \mathbb{R}^{N_2}$, $\mathbf{b}^{(2)}(\omega) \in \mathbb{R}^{M_2}$, $\mathbf{B}(\omega) \in \mathbb{R}^{M_2 \times N_1}$, and $\mathbf{W}(\omega) \in \mathbb{R}^{M_2 \times N_2}$ where random vector $\boldsymbol{\xi}(\omega) = \left[\mathbf{b}'^{(2)}(\omega), \mathbf{c}'^{(2)}(\omega), \mathbf{B}_{1\bullet}(\omega), \ldots, \mathbf{B}_{M_2\bullet}(\omega), \mathbf{W}_{1\bullet}(\omega), \ldots, \mathbf{W}_{M_2\bullet}(\omega)\right]'$.

Matrix $\mathbf{A}$ is referred to as the first period technology matrix. Matrices $\mathbf{B}(\omega)$ are called transition matrices, matrices $\mathbf{W}(\omega)$ are known as recourse matrices, and the vectors $\mathbf{x}^{(2)}(\omega)$ are termed the recourse decision vectors. The problem is said to have *fixed*

recourse when $\mathbf{W}(\omega)$ is not stochastic, i.e. $\mathbf{W}(\omega) = \mathbf{W}$ for fixed $\mathbf{W}$. *Simple recourse* is a special case of fixed recourse with $\mathbf{W} = [\mathbf{I}, -\mathbf{I}]$. The problem has *complete recourse* when for all $\omega \in \Omega$ and for all $\mathbf{x}^{(1)} \in \mathbb{R}^{N_1}$, there exists $\mathbf{x}^{(2)}(\omega) \geq \mathbf{0}$ such that $\mathbf{W}(\omega)\mathbf{x}^{(2)}(\omega) \leq \mathbf{b}^{(2)}(\omega) - \mathbf{B}(\omega)\mathbf{x}^{(1)}$. A special case of complete recourse is *relatively complete recourse*: for all $\omega \in \Omega$ and for all $\mathbf{x}^{(1)} \in \left\{\mathbf{x} \in \mathbb{R}^{N_1} \,\middle|\, \mathbf{Ax} \leq \mathbf{b}^{(1)}, \mathbf{x} \geq \mathbf{0}\right\}$, there exists $\mathbf{x}^{(2)}(\omega) \geq \mathbf{0}$ such that $\mathbf{W}(\omega)\mathbf{x}^{(2)}(\omega) \leq \mathbf{b}^{(2)}(\omega) - \mathbf{B}(\omega)\mathbf{x}^{(1)}$. Note that a problem with simple recourse also has complete recourse.

Problem [2.1] can also be formulated as the so-called *deterministic equivalent problem* (DEP). Define the second period *value function* $\mathfrak{Q}\left[\mathbf{x}^{(1)}, \boldsymbol{\xi}(\omega)\right]$ for given $\mathbf{x}^{(1)}$ and $\boldsymbol{\xi}(\omega)$ as:

$$\begin{aligned} \mathfrak{Q}\left[\mathbf{x}^{(1)}, \boldsymbol{\xi}(\omega)\right] = \max \quad & \mathbf{c}'^{(2)}(\omega)\mathbf{x}^{(2)}(\omega) \\ \text{s.t.} \quad & \mathbf{W}(\omega)\mathbf{x}^{(2)}(\omega) \leq \mathbf{b}^{(2)}(\omega) - \mathbf{B}(\omega)\mathbf{x}^{(1)}, \\ & \mathbf{x}^{(2)}(\omega) \geq \mathbf{0}, \end{aligned}$$

and the second period *expected value function*:

$$\mathcal{Q}\left(\mathbf{x}^{(1)}\right) = \mathcal{E}_{\boldsymbol{\xi}(\omega)}\left(\mathfrak{Q}\left[\mathbf{x}^{(1)}, \boldsymbol{\xi}(\omega)\right]\right).$$

Then the DEP for a two-period stochastic linear program is:

$$\begin{aligned} z^* = \max \quad & \mathbf{c}'^{(1)}\mathbf{x}^{(1)} + \mathcal{Q}\left(\mathbf{x}^{(1)}\right) \\ \text{s.t.} \quad & \mathbf{Ax}^{(1)} \leq \mathbf{b}^{(1)}, \\ & \mathbf{x}^{(1)} \geq \mathbf{0}. \end{aligned} \qquad [2.2]$$

Problem [2.1] is generally difficult to solve. Frauendorfer [80] discusses some of the more common solution procedures for two-period stochastic programs with a general probability measure. Stochastic programs with finitely denumerable probability measures are the most often encountered type of stochastic programs and these are the focus of the remainder of this thesis.

2.1.2 Finitely Denumerable Probability Measures

Problem [2.1] is a linear program (LP) when the universal set Ω contains a finite number of events. The probability measure $\mathcal{P}$ is then a probability mass function over the finite number of subsets in the field $\mathfrak{F}$. A common practice when Ω contains an infinitely denumerable number of events or when the events in Ω are represented by continuous variables is to represent Ω with a finite universal set, say $\widetilde{\Omega}$. The probability measure, $\widetilde{\mathcal{P}}$, on the resulting family of subsets, $\widetilde{\mathfrak{F}}$, is a probability mass function that approximates the probability measure $\mathcal{P}$. This practice dates back to at least 1961 with a paper by Dantzig and Madansky (reference [47]). *Importance sampling* is a recently implemented procedure (see Dempster and Thompson [53], Dantzig and Glynn [45], Dantzig and Infanger [46], Glynn [92], and Infanger [111] and [112]) for variance reduction when $\widetilde{\Omega}$ is created by sampling from Ω. The *stochastic decomposition* procedure of Higle and Sen [97] and [98] (also see Higle, Lowe, and Odio [96]) is a iterative sampling algorithm to obtain an approximate solution to a stochastic program by sequentially taking relatively small discrete samples from Ω.

Let L represent the number of discrete events in $\widetilde{\Omega}$ and let p_l be the probability that event $\widetilde{\omega}_l \in \widetilde{\Omega}$ occurs for $l = 1, \ldots, L$. Further, let the notation $\mathbf{W}^{[l]}$ denote a stochastic array given the occurrence of event $l \in \mathsf{L}$ for indexing set $\mathsf{L} = \{1, \ldots, L\}$, i.e., $\mathbf{W}^{[l]} =$

$\mathbf{W}(\widetilde{\omega}_l)$. Then the stochastic linear program with recourse (SLP) over two periods is:

$$\begin{array}{rrcccl} z^* = \max & \mathbf{c}'^{(1)}\mathbf{x}^{(1)} & + & \sum_{l=1}^{L} p_l \mathbf{c}'^{(2)[l]}\mathbf{x}^{(2)[l]} & & \\ \text{s.t.} & \mathbf{A}\mathbf{x}^{(1)} & & & \leq & \mathbf{b}^{(1)}, \\ & \mathbf{B}^{[l]}\mathbf{x}^{(1)} & + & \mathbf{W}^{[l]}\mathbf{x}^{(2)[l]} & \leq & \mathbf{b}^{(2)[l]}, \quad l = 1, \ldots, L, \\ & & & \mathbf{x}^{(1)} & \geq & \mathbf{0}, \\ & & & \mathbf{x}^{(2)[l]} & \geq & \mathbf{0}, \quad l = 1, \ldots, L. \end{array} \qquad [2.3]$$

Problem [2.3] is a linear program and is generally very large in size. The formulation depicted by problem [2.3] is often called the *extensive* or *grand LP* form of the two-period stochastic linear program – the term grand LP (GLP) will be used hereinafter to describe this formulation. Most solution procedures work with the DEP [2.2] where the discrete second period expected value function is:

$$\mathcal{Q}\left(\mathbf{x}^{(1)}\right) = \sum_{l=1}^{L} p_l \mathfrak{Q}\left[\mathbf{x}^{(1)}, \boldsymbol{\xi}^{[l]}\right], \qquad [2.4]$$

with the discrete second period value function:

$$\begin{array}{rrcl} \mathfrak{Q}\left(\mathbf{x}^{(1)}, \boldsymbol{\xi}^{[l]}\right) = \max & \mathbf{c}'^{(2)[l]}\mathbf{x}^{(2)[l]} & & \\ \text{s.t.} & \mathbf{W}^{[l]}\mathbf{x}^{(2)[l]} & \leq & \mathbf{b}^{(2)[l]} - \mathbf{B}^{[l]}\mathbf{x}^{(1)}, \\ & \mathbf{x}^{(2)[l]} & \geq & \mathbf{0}. \end{array} \qquad [2.5]$$

The two-period SLP forms the foundation for multi-period models. A multi-period SLP can be viewed as a sequence of two-period problems which becomes evident below in the development of the multi-period formulation. A useful design tool for any stochastic linear program is the decision tree and this tool is discussed next.

2.2 Decision Trees and Multi-period Notation

Decision trees, often referred to as event or scenario trees, are extremely useful in developing stochastic linear programs. Figure 2.1 is the decision tree for problem [2.3]. The node (circle) under period 1 is called the *root* node and it denotes the decisions that must be made prior to observing some uncertain event, referred to as an outcome. The nodes under period 2 represent the recourse decisions made in the second period after observing a particular outcome represented by the labeled arcs connecting the first and second period nodes.

Decision trees for multi-period stochastic linear programs and the notational conventions used to navigate through these trees and to describe the corresponding problems are described below.

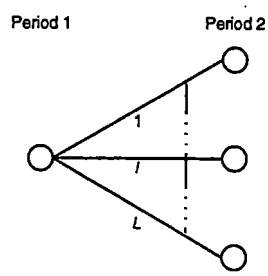


Figure 2.1: Two-Period Decision Tree

2.2.1 Multi-Period Decision Trees

Multi-period stochastic programs extend the *observation-decision* process depicted by the two-period tree shown in Figure 2.1. A set of arcs representing outcomes branches from each node in some period, say t, and each of these arcs terminates in a period $t+1$ node representing the recourse decisions to be made in period $t+1$ given the path of outcomes to that node. The tree extends in this manner through the number of periods, say T, in the planning horizon. Only finite horizon, $T < \infty$, models are examined in this thesis. The terminal nodes in period T are termed *leaf* nodes since no arcs to a following period branch from these nodes. Multi-period decision trees are developed below for two possible stochastic processes – those that terminate with a decision and those that terminate with an observation.

Decision-Terminated Processes

Figure 2.2 illustrates a decision tree for a planning horizon of four periods. Two outcomes of the uncertain event are possible in each period and the stochastic process terminates with a decision represented by a fourth period node. Arcs in the tree are labeled to indicate the index of the outcome represented by the corresponding arc. Nodes are labeled in two ways – a *period-index* format and a *path vector* format. The period-index format is represented by the doublet (t, h_t) of the period number, t (number in the upper half of each node), and the sequential breadth-first (i.e., starting at the top of the tree within each period) index, h_t, of the node (lower half of each node). Path vectors are shown by

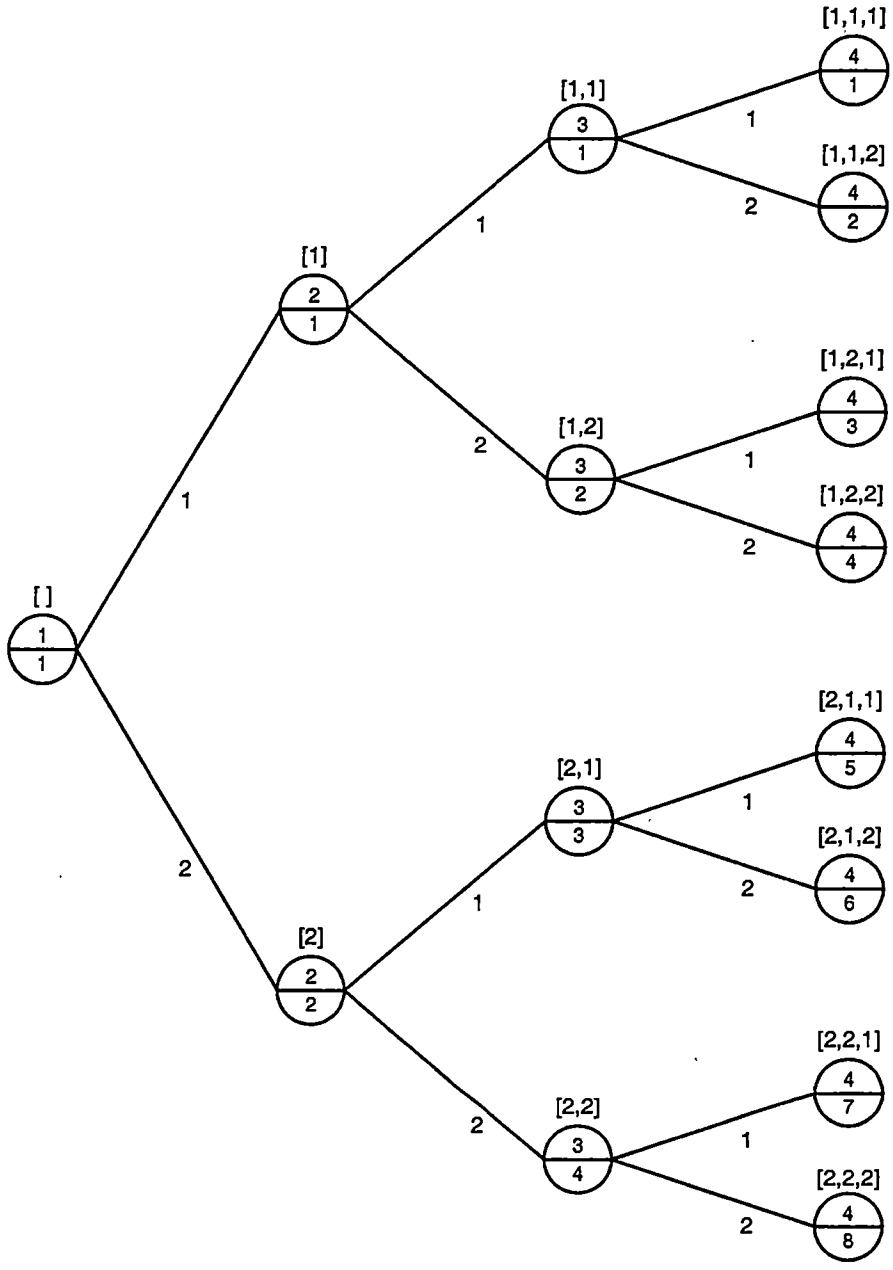


Figure 2.2: Four-Period Decision Tree With Binary Outcomes

the row vectors above each node. The path vector, say $[l_1, l_2, \ldots, l_{t-1}]$, to a node in period t is a row vector of $(t-1)$ elements where element j, $1 \leq j < t$, is the index, l_j, of the outcome in period j along the path of outcomes to the applicable node. A null (empty) vector, $[\,]$, is used to label the single first period root node. The path to a node in period t is called a *t-period scenario* and each T-period scenario is usually simply called a *scenario* of the planning model. There are four 3-period scenarios and eight scenarios in the model depicted by Figure 2.2. Decision trees demonstrate the *nonanticipative* requirement of stochastic programs – decisions in a given period must be made without anticipating future outcomes. For instance, the decisions represented by node (1,1), or equivalently node $[\,]$, are the same regardless of which of the two outcomes is realized in that period or in any following period.

A generic multi-period decision tree is illustrated in Figure 2.3. Only the nodes and arcs in the first and second periods and those along the horizontal path through the root node are labeled. Each node in period t, $t < T$, anchors L_t outcomes so that there are a total of $H_t = \prod_{j=1}^{t-1} L_j$ nodes in period t, $1 \leq t \leq T$, where $H_1 = \prod_{j=1}^{0} L_j = 1$. There are $H^{(t)} = \sum_{j=1}^{t} H_j$ cumulative nodes in periods 1 through $t \leq T$. Note that the convention that all nodes in a given period anchor the same number of outcomes is not a requirement for a stochastic linear program. This convention is adopted because it simplifies notation and involves no loss of generality since each node in a given period may be assigned the same number of outcomes as the node in that period with the maximum number of outcomes

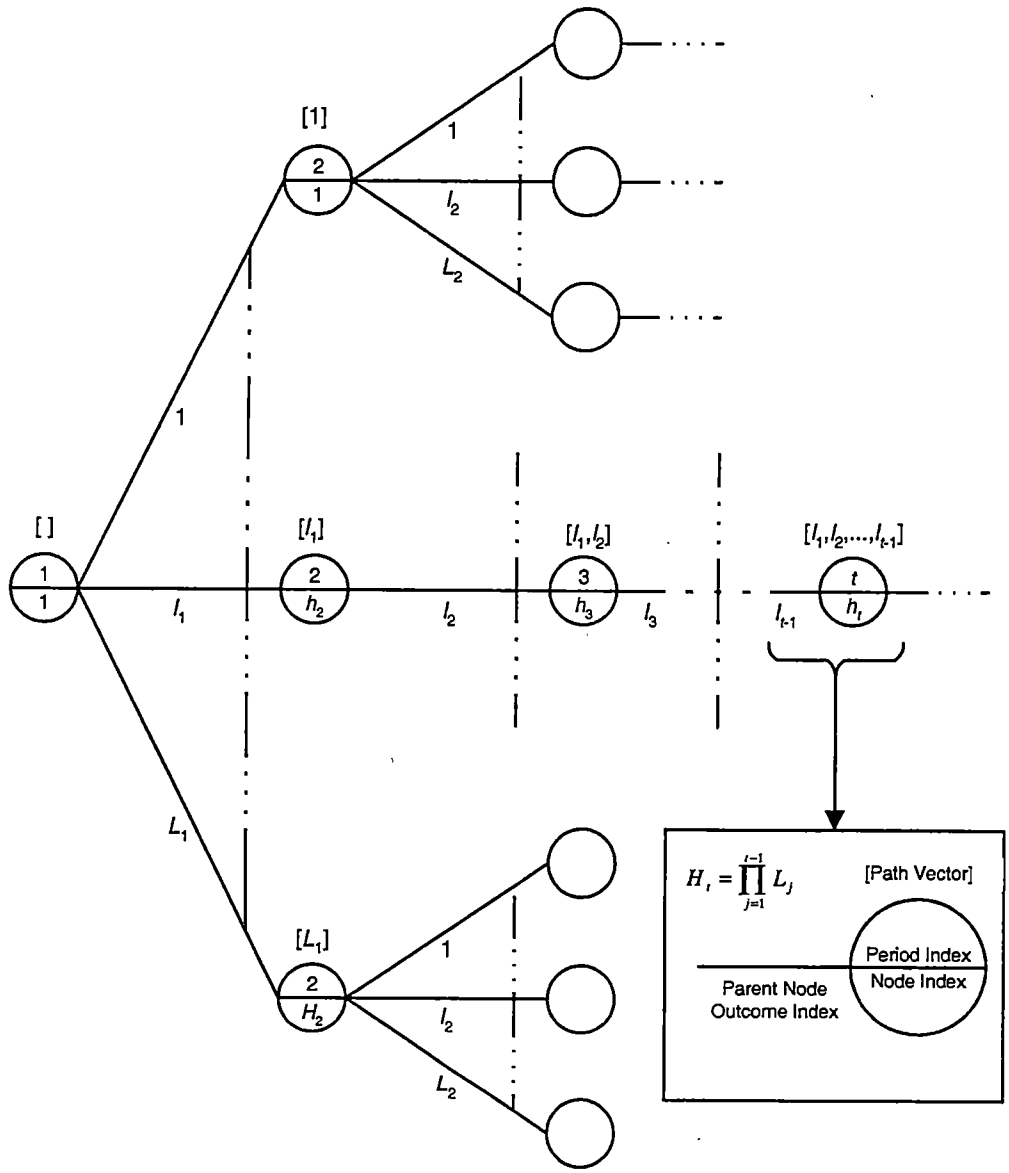


Figure 2.3: Multi-Period Decision Tree

where all excess outcomes are assigned zero probability of occurrence. It is also assumed that $L_t \geq 1$ for $t = 1, \ldots, T-1$.

The period-index and path vector node labeling schemes are equivalent (given the above convention) in the sense that either label can be determined if the other is known. Given the path vector $[l_1, l_2, \ldots, l_{t-1}]$ to a node in period t with $1 \leq l_k \leq L_k$ for $k = 1, \ldots, t-1$, the period-index label for that node is (t, h_t) where

$$h_t = \begin{cases} 1, & \text{if } t = 1, \\ l_{t-1} + \sum_{j=1}^{t-2} (l_j - 1) L_{j+1} L_{j+2} \cdots L_{t-1}, & \text{if } 2 \leq t \leq T. \end{cases} \qquad [2.6]$$

Determining the path vector for a period t node given the period-index label (t, h_t), with $1 \leq h_t \leq H_t$, for that node is more involved than the reverse procedure above. The corresponding path vector is generated by solving equation

$$l_k = 1 + \left\lfloor \frac{h_t - 1 - \sum_{j=1}^{k-1} (l_j - 1) L_{j+1} L_{j+2} \cdots L_{t-1}}{\prod_{j=k+1}^{t-1} L_j} \right\rfloor = \left\lceil \frac{h_t - \sum_{j=1}^{k-1} (l_j - 1) L_{j+1} L_{j+2} \cdots L_{t-1}}{\prod_{j=k+1}^{t-1} L_j} \right\rceil \qquad [2.7]$$

in sequential order for $k = 1, \ldots, t-1$. The operations $\lfloor f \rfloor$ and $\lceil f \rceil$ used in equation [2.7] above are respectively the floor of f (largest integer less than or equal to f) and the ceiling of f (smallest integer greater than or equal to f). Either right-hand-side term in equation [2.7] may be used to determine l_k for $k = 1, \ldots, t-1$. Derivations for equations [2.6] and [2.7] are given in Appendix A.

Observation-Terminated Processes

Each *decision-observation-decision-...-observation-decision* planning process described so far ends in a decision. Some problems may be better represented by a process that terminates with an observation. Prekopa [169, Section 13.1] discusses both the decision-terminated and the observation-terminated processes. Figure 2.4 illustrates a three-period decision tree representing a planning process that terminates with an observation. The arcs emanating from the third period nodes are termed *leafless* outcomes since each represents a possible terminal observation for the process. Each node in the final period of an observation-terminated process must be associated with stochastic decision variables that:

1. are dependent upon the random outcome at that node,
2. are applicable to that node only, and
3. represent an automatic reaction of the process after the outcome has been observed as opposed to an interactive (human) decision made prior to the observation.

These node, or equivalently period, localized stochastic variables are termed *reactive* recourse variables. Variables representing decisions that must be made prior to observing a random outcome are called *discretionary* recourse variables. The three properties of reactive recourse variables are implied by the structure of the decision tree, e.g., Figure 2.4. Reactive recourse variables must exist in the final period since, otherwise, there is no reason to consider the random outcome during that period. They must be applicable only to the final period node in question since there is no future period. Finally, reactive

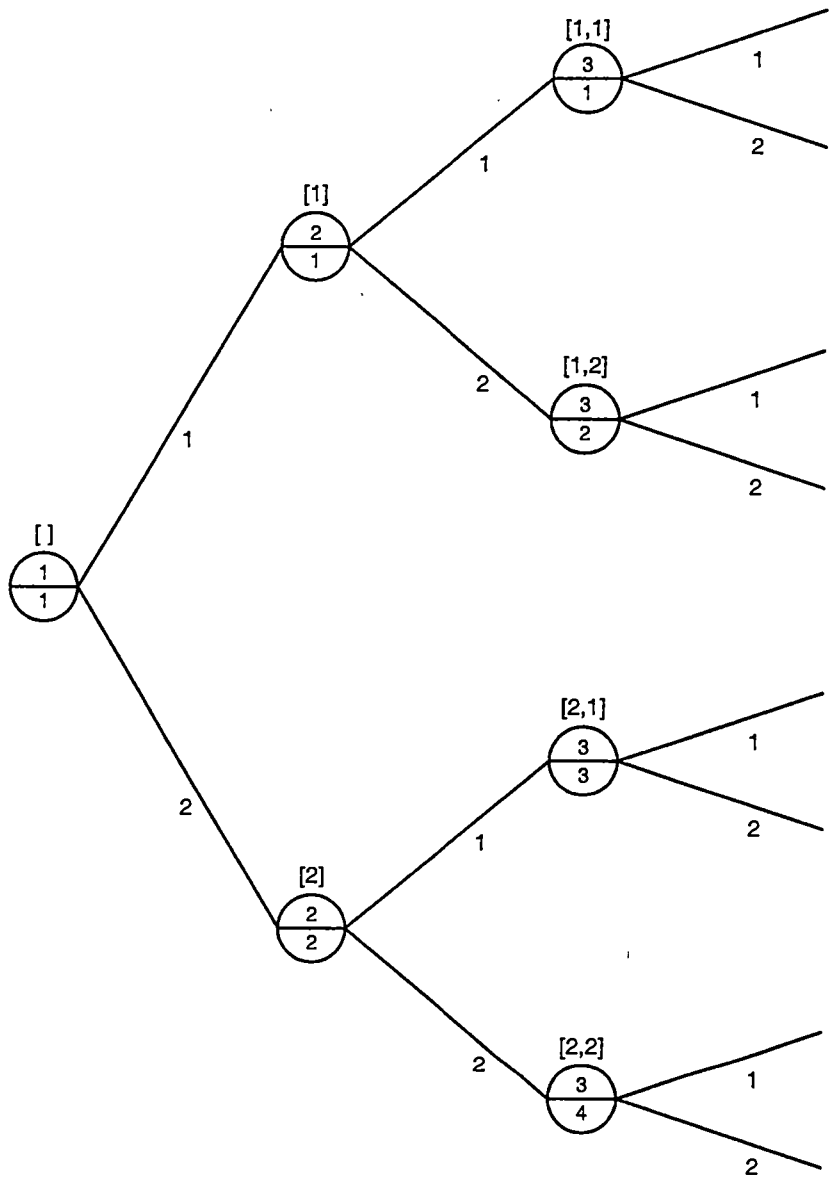


Figure 2.4: Three-Period Decision Tree Ending in an Observation

recourse variables cannot represent a pre-observation decision since this would violate the nonanticipative requirement of the stochastic program.

Reactive recourse variables are not restricted to the final period – they may be present in all periods, including the first. Observation-terminated processes require some modifications to the definitions of the problem arrays and to the notational conventions developed for decision-terminated processes. These modifications are minor and are discussed in detail in Chapter 5 with the development of an observation-terminated market investment model.

Additional notational conventions used for multi-period trees and stochastic linear programs are introduced and defined in the next subsection.

2.2.2 Multi-period Notation

Nodes in adjacent periods of a decision tree are said to have a *parent-child* relationship. Each node in period t, $t < T$, of Figure 2.3 is the parent of L_t child nodes in period $t+1$ and each node in period t, $t > 1$, is the child of a parent node in period $t-1$. Nodes in period t with the same parent node are termed *siblings* and nodes in period t with different parents are termed *cousins*. Nodes to the left of (i.e., in an earlier period) or above a given node are said to be *older* than that node.

Let the row vector $\left[\tilde{l}_1, \ldots, \tilde{l}_{t-1}\right]$ be the path vector to a node in period t. The set $\mathsf{H}^{\mathsf{P}}_{t-1}\left(\left[\tilde{l}_1, \ldots, \tilde{l}_{t-1}\right]\right)$ containing the corresponding unique parent node in period $t-1$ is

then defined as:

$$\mathsf{H}^{\mathsf{P}}_{t-1}\left(\left[\tilde{l}_1,\ldots,\tilde{l}_{t-1}\right]\right) = \begin{cases} \emptyset, & \text{if } t=1, \\ \left\{\left[\tilde{l}_1,\ldots,\tilde{l}_{t-2}\right]\right\}, & \text{if } 2 \le t \le T, \end{cases}$$

where $\mathsf{H}^{\mathsf{P}}_1\left(\left[\tilde{l}_1\right]\right) = \left\{\left[\tilde{l}_0\right]\right\} = \{[\,]\}$, i.e., the parent node of a second period node is the single first period node. Similarly, the set $\mathsf{H}^{\mathsf{C}}_{t+1}\left(\left[\tilde{l}_1,\ldots,\tilde{l}_{t-1}\right]\right)$ containing the period $t+1$ child nodes of node $\left[\tilde{l}_1,\ldots,\tilde{l}_{t-1}\right]$ is defined as:

$$\mathsf{H}^{\mathsf{C}}_{t+1}\left(\left[\tilde{l}_1,\ldots,\tilde{l}_{t-1}\right]\right) = \begin{cases} \left\{\left[\tilde{l}_1,\ldots,\tilde{l}_{t-1},l_t\right] \,\middle|\, l_t = 1,\ldots,L_t\right\}, & \text{if } 1 \le t < T, \\ \emptyset, & \text{if } t = T. \end{cases}$$

The two definitions above demonstrate a major advantage that the path vector node labeling scheme has over the period-index scheme – parent-child relationships are obvious in the path vector scheme while such relationships are not easily discernible with the period-index scheme. For instance, calculations involving the number of outcomes in periods 1 through t would be required to determine if node (t, h_t) is the parent of node $(t+1, h_{t+1})$. A disadvantage of the path vector scheme is that the size of the vector grows with the depth of the node in the tree. Identifying problem arrays that correspond to a particular node by labeling the arrays with the path vector for that node can therefore be cumbersome and impractical. This disadvantage can be alleviated by adopting the shorthand notation that $[\bullet]_t$ represents the path vector for a period t node. Path vector $[\bullet]_t$ then has $(t-1)$ elements where $[\bullet]_1 = [\,]$ is the single first period root node. Then, by convention,

$$[\bullet]_{t+1} \in \mathsf{H}^{\mathsf{C}}_{t+1}\left([\bullet]_t\right) \iff [\bullet]_t \in \mathsf{H}^{\mathsf{P}}_t\left([\bullet]_{t+1}\right), \text{ and}$$

$$[\bullet]_t \in \mathsf{H}^{\mathsf{C}}_t\left([\bullet]_{t-1}\right) \iff [\bullet]_{t-1} \in \mathsf{H}^{\mathsf{P}}_{t-1}\left([\bullet]_t\right),$$

when $[\bullet]_t$ and $[\bullet]_{t-1}$ or $[\bullet]_{t+1}$ are used in the same expression. Further, an expression like

$$\sum_{l_t=1}^{L_t} \mathcal{O}\left(\mathbf{x}^{[\bullet]_t}, \boldsymbol{\xi}^{[\bullet]_{t+1}}\right)$$

for operator $\mathcal{O}(\bullet)$ and operands $\mathbf{x}^{[\bullet]_t}$ and $\boldsymbol{\xi}^{[\bullet]_{t+1}}$ where $[\bullet]_t = \left[\tilde{l}_1, \ldots, \tilde{l}_{t-1}\right]$ implies the operation:

$$\sum_{l_t=1}^{L_t} \mathcal{O}\left(\mathbf{x}^{[\bullet]_t}, \boldsymbol{\xi}^{[\bullet]_{t+1}}\right) = \mathcal{O}\left(\mathbf{x}^{[\tilde{l}_1,\ldots,\tilde{l}_{t-1}]}, \boldsymbol{\xi}^{[\tilde{l}_1,\ldots,\tilde{l}_{t-1},1]}\right) + \ldots + \mathcal{O}\left(\mathbf{x}^{[\tilde{l}_1,\ldots,\tilde{l}_{t-1}]}, \boldsymbol{\xi}^{[\tilde{l}_1,\ldots,\tilde{l}_{t-1},L_t]}\right).$$

The shorthand $[\bullet]_t$ notation is also used to implicitly indicate the dependence of a stochastic array in period t on the occurrence of the outcomes implied by $[\bullet]_t$ in periods 1 through $t-1$. Let $\omega_{l_t}^{(t)}$ represent the period t outcome with index l_t, $1 \leq l_t \leq L_t$, and let $[\bullet]_t = \left[\tilde{l}_1, \ldots, \tilde{l}_{t-1}\right]$ represent a node in period t, $1 \leq t \leq T$. Then, for example,

$$\mathbf{W}^{[\bullet]_t} = \mathbf{W}\left(\omega_{\tilde{l}_{t-1}}^{(t-1)} \middle| \omega_{\tilde{l}_{t-2}}^{(t-2)}, \ldots, \omega_{\tilde{l}_2}^{(2)}, \omega_{\tilde{l}_1}^{(1)}\right),$$

represents the conditional recourse matrix at node $[\bullet]_t$.

The conditional probability that the outcome with index l_t, $1 \leq l_t \leq L_t$, $1 \leq t \leq T-1$, is observed at the t-period node $[\bullet]_t$ given the indicated outcomes $\tilde{l}_1, \ldots, \tilde{l}_{t-1}$ in periods 1 through $t-1$ is $p_{l_t}^{[\bullet]_t}$, i.e.,

$$p_{l_t}^{[\bullet]_t} = \mathfrak{P}\left(\omega_{l_t}^{(t)} \middle| \omega_{\tilde{l}_{t-1}}^{(t-1)}, \omega_{\tilde{l}_{t-2}}^{(t-2)}, \ldots, \omega_{\tilde{l}_2}^{(2)}, \omega_{\tilde{l}_1}^{(1)}\right),$$

where $\mathfrak{P}(\bullet)$ is the probability operator and $p_{l_1}^{[\bullet]_1} = \mathfrak{P}\left(\omega_{l_1}^{(1)} \middle| \omega_0^{(0)}\right) = \mathfrak{P}\left(\omega_{l_1}^{(1)}\right) = p_{l_1}^{[]}$. The compound (or, joint) probability, $\hat{p}^{[\bullet]_t}$, that the process enters the period t node $[\bullet]_t = \left[\tilde{l}_1, \ldots, \tilde{l}_{t-1}\right]$, $1 \leq t \leq T$, is the product of the conditional probabilities of the outcomes

along the indicated path to that node:

$$\hat{p}^{[\bullet]_t} = \hat{p}^{[\tilde{l}_1,\tilde{l}_2,\ldots,\tilde{l}_{t-2},\tilde{l}_{t-1}]} = p_{\tilde{l}_1}^{[]} p_{\tilde{l}_2}^{[\tilde{l}_1]} \cdots p_{\tilde{l}_{t-1}}^{[\tilde{l}_1,\tilde{l}_2,\ldots,\tilde{l}_{t-2}]} = \prod_{j=1}^{t-1} p_{\tilde{l}_j}^{[\bullet]_j},$$

where $\hat{p}^{[\bullet]_1} = \hat{p}^{[]} = 1$, i.e., the single first period node is always entered.

Either labeling scheme may be used to label problem arrays, e.g., $\mathbf{x}^{[l_1,\ldots,l_{t-1}]} = \mathbf{x}^{[\bullet]_t} = \mathbf{x}^{(t,h_t)}$. The path vector scheme, especially with the shorthand $[\bullet]_t$ notation, is used most often. A combination of the two schemes will sometimes be used to improve clarity and explanations for such cases are always provided. The notational conventions established above are used below to develop the formulation for a multi-period stochastic linear program.

2.3 Multi-Period Stochastic Linear Programs

Formulations for a stochastic linear program with recourse over a planning horizon divided into multiple periods can now be defined. The first formulation below is referred to as the primal formulation of the multi-period problem and the second formulation is the dual to the primal formulation. Formulation development is followed by a discussion comparing the two formulations.

2.3.1 Primal Formulation

The primal formulation of the multi-period stochastic linear program is an extension to the two-period problem [2.3]. The following grand LP formulation is referred to as problem

PMPGLP:

$$
\begin{aligned}
z^{\mathrm{P}} = \max \;\; & \mathbf{c}'^{[]}\mathbf{x}^{[]} + \sum_{l_1=1}^{L_1} p_{l_1}^{[]} \left[\mathbf{c}'^{[\bullet]_2}\mathbf{x}^{[\bullet]_2} + \sum_{l_2=1}^{L_2} p_{l_2}^{[\bullet]_2} \left[\mathbf{c}'^{[\bullet]_3}\mathbf{x}^{[\bullet]_3} + \ldots + \right.\right. \\
& \left.\left. \sum_{l_{T-1}=1}^{L_{T-1}} p_{l_{T-1}}^{[\bullet]_{T-1}} \left[\mathbf{c}'^{[\bullet]_T}\mathbf{x}^{[\bullet]_T}\right] \ldots\right]\right] \\
\text{s.t.} \quad & \mathbf{A}\mathbf{x}^{[]} \leq \mathbf{b}^{[]}, \\
& \mathbf{B}^{[\bullet]_t}\mathbf{x}^{[\bullet]_{t-1}} + \mathbf{W}^{[\bullet]_t}\mathbf{x}^{[\bullet]_t} \leq \mathbf{b}^{[\bullet]_t}, \quad h_t = 1, \ldots, H_t, \quad t = 2, \ldots, T, \\
& \mathbf{x}^{[]} \geq \mathbf{0}, \\
& \mathbf{x}^{[\bullet]_t} \geq \mathbf{0}, \quad h_t = 1, \ldots, H_t, \quad t = 2, \ldots, T,
\end{aligned}
\qquad [2.8]
$$

where equation [2.7] is used to determine $[\bullet]_t$ given (t, h_t) in the second and fourth sets of constraints. The coefficient matrices in the first two sets of constraints above form a special *staircase* structure that is difficult to demonstrate for the general problem due to the sheer size of the problem. Composite matrices formed by separately combining all transition matrices and then all recourse matrices for a period into two matrices for that period will be used for the demonstration. Let $\mathbf{x}_t$, $t = 2, \ldots, T$, be the vector formed by stacking the recourse decision vectors for each node in period t in breadth-first node order, i.e.,

$$
\mathbf{x}_t = \begin{bmatrix} \mathbf{x}^{(t,1)} \\ \vdots \\ \mathbf{x}^{(t,h_t)} \\ \vdots \\ \mathbf{x}^{(t,H_t)} \end{bmatrix} = \begin{bmatrix} \mathbf{x}^{[1,1,\ldots,1,1]} \\ \vdots \\ \mathbf{x}^{[l_1,l_2,\ldots,l_{t-2},l_{t-1}]} \\ \vdots \\ \mathbf{x}^{[L_1,L_2,\ldots,L_{t-2},L_{t-1}]} \end{bmatrix},
$$

and let $\mathbf{x}_1 = \mathbf{x}^{[]}$. Define $\mathbf{B}_t$ as the matrix obtained by stacking the transition matrices for each node in period t in breadth-first node order, in columnar blocks to conform with $\mathbf{x}_{t-1}$,

and with empty blocks filled with zeros. Then, for example, $\mathbf{B}_3\mathbf{x}_2$ will be

$$\mathbf{B}_3\mathbf{x}_2 = \begin{bmatrix} \mathbf{B}^{[1,1]} & & & & \\ \vdots & & & & \\ \mathbf{B}^{[1,L_2]} & & & & \\ & \mathbf{B}^{[2,1]} & & & \\ & \vdots & & & \\ & \mathbf{B}^{[2,L_2]} & & & \\ & & \ddots & & \\ & & & \mathbf{B}^{[L_1,1]} & \\ & & & \vdots & \\ & & & \mathbf{B}^{[L_1,L_2]} & \end{bmatrix} \begin{bmatrix} \mathbf{x}^{[1]} \\ \mathbf{x}^{[2]} \\ \vdots \\ \mathbf{x}^{[L_1]} \end{bmatrix},$$

where the empty blocks in $\mathbf{B}_3$ above are appropriately sized zero-matrices. Similarly, let $\mathbf{W}_t$ be the block-diagonal matrix formed by combining the recourse matrices in period t so that $\mathbf{W}_3\mathbf{x}_3$ will be

$$\mathbf{W}_3\mathbf{x}_3 = \begin{bmatrix} \mathbf{W}^{[1,1]} & & & & & \\ & \ddots & & & & \\ & & \mathbf{W}^{[1,L_2]} & & & \\ & & & \ddots & & \\ & & & & \mathbf{W}^{[L_1,1]} & \\ & & & & & \ddots \\ & & & & & \mathbf{W}^{[L_1,L_2]} \end{bmatrix} \begin{bmatrix} \mathbf{x}^{[1,1]} \\ \vdots \\ \mathbf{x}^{[1,L_2]} \\ \vdots \\ \mathbf{x}^{[L_1,1]} \\ \vdots \\ \mathbf{x}^{[L_1,L_2]} \end{bmatrix}.$$

Figure 2.5 then illustrates the staircase structure formed by the composite coefficient matrices of the first two sets of constraints in problem [2.8].

The multi-period deterministic equivalent problem for the primal formulation is created by defining value functions and expected value functions for each period after the first in the planning horizon. Define the period t expected value function, $\mathcal{Q}_t\left(\mathbf{x}^{[\bullet]_{t-1}}\right)$, for

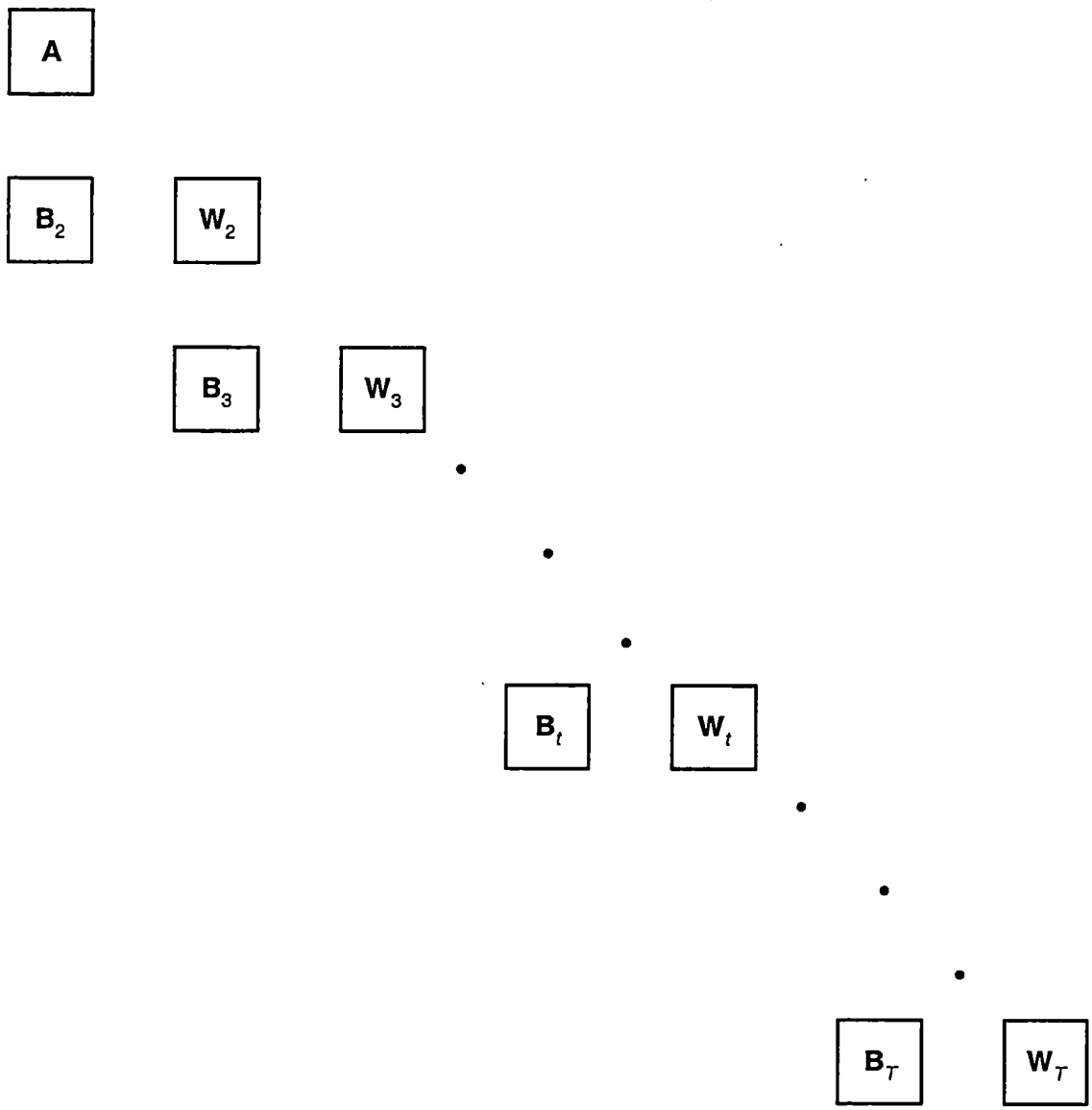


Figure 2.5: Staircase Structure of Problem **PMPGLP** [2.8]

$t = 2, \ldots, T$ as

$$\mathcal{Q}_t\left(\mathbf{x}^{[\bullet]_{t-1}}\right) = \sum_{l_{t-1}=1}^{L_{t-1}} p_{l_{t-1}}^{[\bullet]_{t-1}} \mathfrak{Q}_t\left(\mathbf{x}^{[\bullet]_{t-1}}, \boldsymbol{\xi}^{[\bullet]_t}\right),$$

where the period T value function, $\mathfrak{Q}_T\left(\mathbf{x}^{[\bullet]_{T-1}}, \boldsymbol{\xi}^{[\bullet]_T}\right)$, is defined as

$$\begin{aligned} \mathfrak{Q}_T\left(\mathbf{x}^{[\bullet]_{T-1}}, \boldsymbol{\xi}^{[\bullet]_T}\right) = \max \quad & \mathbf{c}'^{[\bullet]_T}\mathbf{x}^{[\bullet]_T} \\ \text{s.t.} \quad & \mathbf{W}^{[\bullet]_T}\mathbf{x}^{[\bullet]_T} \leq \mathbf{b}^{[\bullet]_T} - \mathbf{B}^{[\bullet]_T}\mathbf{x}^{[\bullet]_{T-1}}, \\ & \mathbf{x}^{[\bullet]_T} \geq \mathbf{0}, \end{aligned}$$

and the period t value function, $\mathfrak{Q}_t\left(\mathbf{x}^{[\bullet]_{t-1}}, \boldsymbol{\xi}^{[\bullet]_t}\right)$, for $t = 2, \ldots, T-1$ is

$$\begin{aligned} \mathfrak{Q}_t\left(\mathbf{x}^{[\bullet]_{t-1}}, \boldsymbol{\xi}^{[\bullet]_t}\right) = \max \quad & \mathbf{c}'^{[\bullet]_t}\mathbf{x}^{[\bullet]_t} + \mathcal{Q}_{t+1}\left(\mathbf{x}^{[\bullet]_t}\right) \\ \text{s.t.} \quad & \mathbf{W}^{[\bullet]_t}\mathbf{x}^{[\bullet]_t} \leq \mathbf{b}^{[\bullet]_t} - \mathbf{B}^{[\bullet]_t}\mathbf{x}^{[\bullet]_{t-1}}, \\ & \mathbf{x}^{[\bullet]_t} \geq \mathbf{0}. \end{aligned}$$

Then the multi-period DEP is

$$\begin{aligned} z^{\mathrm{P}} = \max \quad & \mathbf{c}'^{[]}\mathbf{x}^{[]} + \mathcal{Q}_2\left(\mathbf{x}^{[]}\right) \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x}^{[]} \leq \mathbf{b}^{[]}, \\ & \mathbf{x}^{[]} \geq \mathbf{0}. \end{aligned}$$

The DEP then resembles a sequence of two-period problems due to the recursive nature of the period t value function for $t = 2, \ldots, T-1$.

The deterministic equivalent problem may be reformulated by redefining the value functions to remove the functional dependence on the expected value functions. A value function, referred to as a *nodal value function*, can be defined for each node in periods after the first in the decision tree. The nodal value function for a period T node is

$$\begin{aligned} \mathfrak{Q}^{[\bullet]_T}\left(\mathbf{x}^{[\bullet]_{T-1}}\right) = p_{l_{T-1}}^{[\bullet]_{T-1}} \max \quad & \mathbf{c}'^{[\bullet]_T}\mathbf{x}^{[\bullet]_T} \\ \text{s.t.} \quad & \mathbf{W}^{[\bullet]_T}\mathbf{x}^{[\bullet]_T} \leq \mathbf{b}^{[\bullet]_T} - \mathbf{B}^{[\bullet]_T}\mathbf{x}^{[\bullet]_{T-1}}, \\ & \mathbf{x}^{[\bullet]_T} \geq \mathbf{0}, \end{aligned} \qquad [2.9]$$

and the function for a node in period t, $2 \leq t \leq T-1$, is

$$\begin{aligned} \mathfrak{Q}^{[\bullet]_t}\left(\mathbf{x}^{[\bullet]_{t-1}}\right) = p_{l_{t-1}}^{[\bullet]_{t-1}} \max \quad & \mathbf{c}'^{[\bullet]_t}\mathbf{x}^{[\bullet]_t} + \sum_{l_t=1}^{L_t} \mathfrak{Q}^{[\bullet]_{t+1}}\left(\mathbf{x}^{[\bullet]_t}\right) \\ \text{s.t.} \quad & \mathbf{W}^{[\bullet]_t}\mathbf{x}^{[\bullet]_t} \leq \mathbf{b}^{[\bullet]_t} - \mathbf{B}^{[\bullet]_t}\mathbf{x}^{[\bullet]_{t-1}}, \\ & \mathbf{x}^{[\bullet]_t} \geq \mathbf{0}. \end{aligned} \qquad [2.10]$$

Note that no random component argument, $\boldsymbol{\xi}^{[\bullet]_t}$, is required since the stochastic data are treated as being assigned to the nodes of the decision tree. The multi-period DEP, referred to as problem **PMPDEP**, is then

$$\begin{aligned} z^{\mathrm{P}} = \max \mathbf{c}'^{[]}\mathbf{x}^{[]} + \sum_{l_1=1}^{L_1} \mathfrak{Q}^{[\bullet]_2}\left(\mathbf{x}^{[]}\right) \\ \text{s.t.} \quad \mathbf{A}\mathbf{x}^{[]} \leq \mathbf{b}^{[]}, \\ \mathbf{x}^{[]} \geq \mathbf{0}. \end{aligned} \qquad [2.11]$$

2.3.2 Dual Formulation

The dual formulation to problem **PMPGLP** [2.8] is called problem **DMPGLP**:

$$z^{\mathrm{D}} = \min \mathbf{b}'^{[]}\boldsymbol{\pi}^{[]} + \sum_{l_1}^{L_1} \left[\mathbf{b}'^{[\bullet]_2}\boldsymbol{\pi}^{[\bullet]_2} + \left[\sum_{l_2}^{L_2} \mathbf{b}'^{[\bullet]_3}\boldsymbol{\pi}^{[\bullet]_3} + \left[\ldots + \left[\sum_{l_{T-1}}^{L_{T-1}} \mathbf{b}'^{[\bullet]_T}\boldsymbol{\pi}^{[\bullet]_T} \right] \ldots \right] \right] \right]$$

$$\begin{aligned} \text{s.t.} \quad \mathbf{A}'\boldsymbol{\pi}^{[]} + \sum_{l_1=1}^{L_1} \mathbf{B}'^{[\bullet]_2}\boldsymbol{\pi}^{[\bullet]_2} &\geq \mathbf{c}^{[]}, \\ \mathbf{W}'^{[\bullet]_t}\boldsymbol{\pi}^{[\bullet]_t} + \sum_{l_t=1}^{L_t} \mathbf{B}'^{[\bullet]_{t+1}}\boldsymbol{\pi}^{[\bullet]_{t+1}} &\geq \hat{p}^{[\bullet]_t}\mathbf{c}^{[\bullet]_t}, \quad h_t = 1, \ldots, H_t, \; t = 2, \ldots, T-1, \\ \mathbf{W}'^{[\bullet]_T}\boldsymbol{\pi}^{[\bullet]_T} &\geq \hat{p}^{[\bullet]_T}\mathbf{c}^{[\bullet]_T}, \quad h_T = 1, \ldots, H_T, \\ \boldsymbol{\pi}^{[]} &\geq \mathbf{0}, \\ \boldsymbol{\pi}^{[\bullet]_t} &\geq \mathbf{0}, \quad h_t = 1, \ldots, H_t, \; t = 2, \ldots, T, \end{aligned} \qquad [2.12]$$

where equation [2.7] is used to determine $[\bullet]_t$ given (t, h_t) in the applicable constraints. The constraints of problem **DMPGLP**, like those of problem **PMPGLP** [2.8], also exhibit a staircase structure. Figure 2.6 shows the staircase structure of problem **DMPGLP** using the composite matrix notation defined above.

A deterministic equivalent problem for the dual formulation is not defined since one is not used in any of the solution procedures to be developed in Chapter 4. These procedures will directly exploit the staircase structure of problem **DMPGLP** [2.12].

2.3.3 Comparing Formulations

Figures 2.5 on page 37 and 2.6 on page 41 demonstrate both the major similarity between the primal and dual formulations and the major difference between the formulations. Both formulations exhibit staircase structures but the two structures differ significantly. The primal staircase structure for problem **PMPGLP** [2.8], Figure 2.5, is *transition-supported* since the steps, the composite recourse (**W**) matrices, are supported by the transition (B) matrices. On the other hand, the dual staircase structure for problem **DMPGLP** [2.12] is *recourse-supported* since the recourse matrices provide support for the transition matrices. This structural difference results in diametrical routes for passing information in the two structures. Problem **PMPGLP** sends information (the primal $\mathbf{x}$ variables) forward from a period to the next period, whereas problem **DMPGLP** sends information (the dual π variables) backward from a period to the preceding period. These concepts will be applied

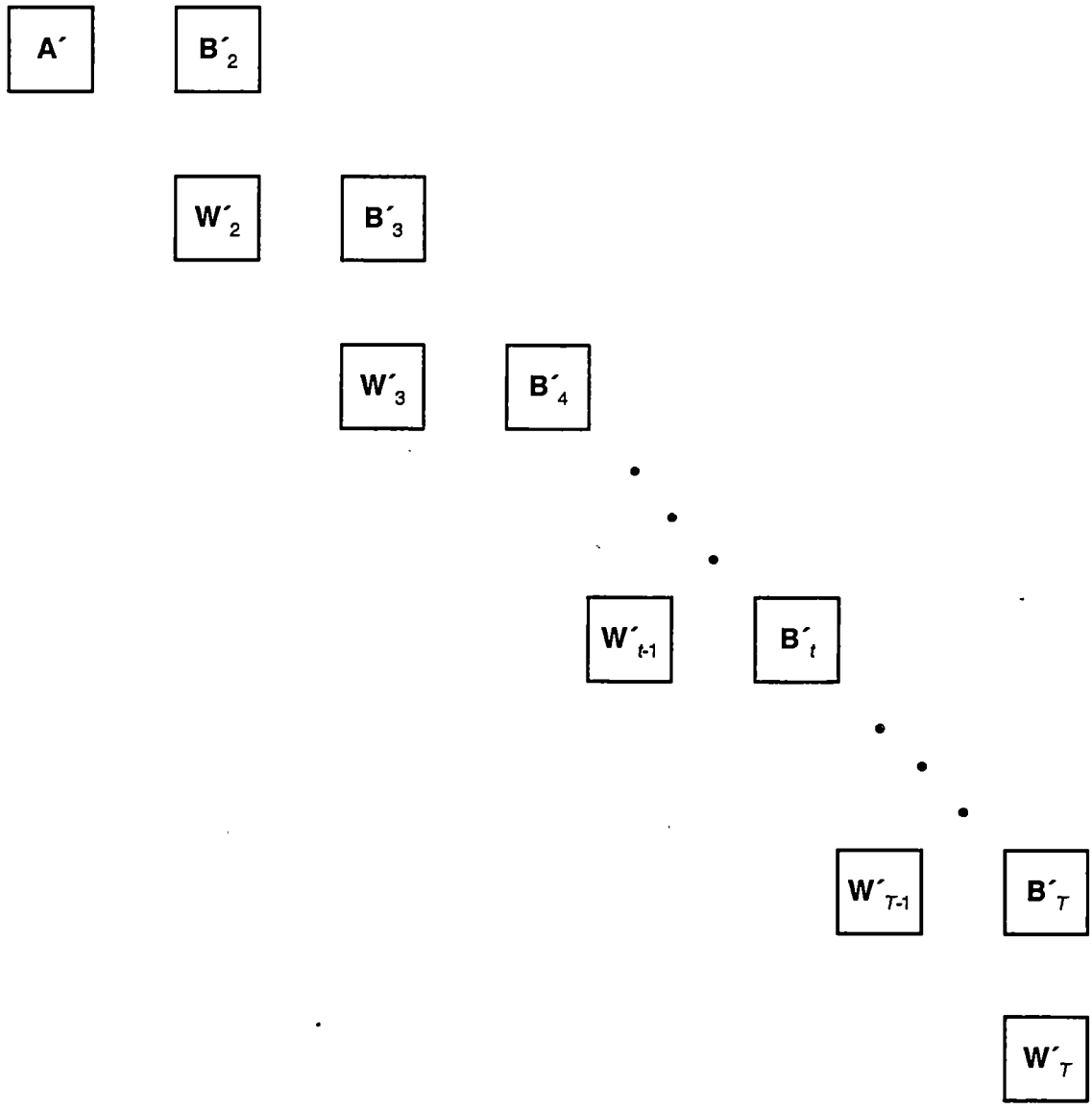


Figure 2.6: Staircase Structure of Problem **DMPGLP** [2.12]

in Chapter 4 to develop decomposition procedures to obtain solutions to a stochastic linear program.

2.4 Block-Separable Recourse

Block-separable recourse is a property of some stochastic programs that permits the problem structure to be manipulated in many ways. This property was first described by Louveaux [134] and is also discussed in Birge and Louveaux [16, Section 3.5]. Let [0] represent an appropriately sized zero-matrix, then a stochastic linear program is said to have block-separable recourse when the first period technology matrix and all transition and recourse matrices can be partitioned as follows:

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}^{(1,1)} & [0] \\ \mathbf{A}^{(2,1)} & \mathbf{A}^{(2,2)} \end{bmatrix},$$

$$\mathbf{B}^{[\bullet]_t} = \begin{bmatrix} \mathbf{B}^{(1,1)[\bullet]_t} & [0] \\ \mathbf{B}^{(2,1)[\bullet]_t} & [0] \end{bmatrix} \text{ and } \mathbf{W}^{[\bullet]_t} = \begin{bmatrix} \mathbf{W}^{(1,1)[\bullet]_t} & [0] \\ \mathbf{W}^{(2,1)[\bullet]_t} & \mathbf{W}^{(2,2)[\bullet]_t} \end{bmatrix} \text{ for } \begin{cases} h_t = 1, \ldots, H_t, \\ t = 2, \ldots, T, \end{cases}$$

Equation [2.7] is used to determine $[\bullet]_t$ given (t, h_t) in the lower sets of partitions above and where appropriate in the discussion that follows. Note that these partitions differ from those defined in the two references above since those authors stipulate that $\mathbf{W}^{(2,1)[\bullet]_t} = \mathbf{0}$ for $t = 2, \ldots, T$. The partitioning of $\mathbf{W}^{[\bullet]_t}$ defined above is more general while still allowing block-separability as shown below.

Problem vectors are then partitioned accordingly:

$$\mathbf{b}^{[\bullet]_t} = \begin{bmatrix} \mathbf{b}^{(1)[\bullet]_t} \\ \mathbf{b}^{(2)[\bullet]_t} \end{bmatrix}, \mathbf{c}^{[\bullet]_t} = \begin{bmatrix} \mathbf{c}^{(1)[\bullet]_t} \\ \mathbf{c}^{(2)[\bullet]_t} \end{bmatrix}, \text{ and } \mathbf{x}^{[\bullet]_t} = \begin{bmatrix} \mathbf{x}^{(1)[\bullet]_t} \\ \mathbf{x}^{(2)[\bullet]_t} \end{bmatrix} \text{ for } \begin{cases} h_t = 1, \ldots, H_t, \\ t = 1, \ldots, T, \end{cases}$$

Louveaux calls the $\mathbf{x}^{(1)[\bullet]_t}$ variables *aggregate* level decision variables and the $\mathbf{x}^{(2)[\bullet]_t}$ variables *detailed* level decision variables.

Constraints in the primal problem **PMPGLP** [2.8] may then be written as:

$$\mathbf{A}\mathbf{x}^{[]} \leq \mathbf{b}^{[]} \Leftrightarrow \begin{cases} \mathbf{A}^{(1,1)}\mathbf{x}^{(1)[]} & \leq \mathbf{b}^{(1)[]} \\ \mathbf{A}^{(2,1)}\mathbf{x}^{(1)[]} + \mathbf{A}^{(2,2)}\mathbf{x}^{(2)[]} & \leq \mathbf{b}^{(2)[]} \end{cases},$$

and

$$\mathbf{B}^{[\bullet]_t}\mathbf{x}^{[\bullet]_{t-1}} + \mathbf{W}^{[\bullet]_t}\mathbf{x}^{[\bullet]_t} \leq \mathbf{b}^{[\bullet]_t} \Leftrightarrow$$

$$\begin{cases} \mathbf{B}^{(1,1)[\bullet]_t}\mathbf{x}^{(1)[\bullet]_{t-1}} + \mathbf{W}^{(1,1)[\bullet]_t}\mathbf{x}^{(1)[\bullet]_t} & \leq \mathbf{b}^{(1)[\bullet]_t} \\ \mathbf{B}^{(2,1)[\bullet]_t}\mathbf{x}^{(1)[\bullet]_{t-1}} + \mathbf{W}^{(2,1)[\bullet]_t}\mathbf{x}^{(1)[\bullet]_t} + \mathbf{W}^{(2,2)[\bullet]_t}\mathbf{x}^{(2)[\bullet]_t} & \leq \mathbf{b}^{(2)[\bullet]_t} \end{cases} \text{ for } \begin{cases} h_t = 1, \ldots, H_t, \\ t = 2, \ldots, T. \end{cases}$$

Note that the detailed level decision variables at a node in period $t-1$ have no effect on the constraints in period t for $t = 2, \ldots, T$ due to the partitioning of the transition matrices. This characteristic of block-separable programs allows the constraints to be arranged in many different ways that may prove beneficial in finding a solution. For instance, all constraints at nodes in periods after the first that incorporate only aggregate level decision variables, i.e.,

$$\mathbf{B}^{(1,1)[\bullet]_t}\mathbf{x}^{(1)[\bullet]_{t-1}} + \mathbf{W}^{(1,1)[\bullet]_t}\mathbf{x}^{(1)[\bullet]_t} \leq \mathbf{b}^{(1)[\bullet]_t},\ h_t = 1, \ldots, H_t,\ t = 2, \ldots, T,$$

can be moved to directly follow the first period constraints $\mathbf{A}^{(1,1)}\mathbf{x}^{(1)[]} \leq \mathbf{b}^{(1)[]}$. Then, since the objective function can be separated accordingly, the rearranged problem structure will look exactly like a two-period problem. All aggregate level decision variables appear in the 'first' period and the detailed level decision variables appear only in the 'second' period. Similar rearrangements can be performed with the dual formulation. This and other con-

straint groupings are discussed in detail in Chapter 5 with the application of decomposition to a multi-period market investment model.

Block-separable recourse is fairly common in stochastic programming. Most multi-period stochastic programs are inherently block-separable or can be induced to be block-separable according to Louveaux [134, page 48]. An example of the latter case is observation-terminated processes with reactive recourse variables in all periods – see the last subsection of Section 2.2.1 above. Reactive recourse variables correspond to the detailed level decision variables while discretionary recourse variables correspond to the aggregate level decision variables. Transition matrices in this type of problem are clearly partitioned as above since no reactive recourse variables are passed from one period to the next. Recourse matrices and the first period technology matrix can be induced to have the above partitions when necessary by adding artificial constraints. Assume, for example, that the matrix $\mathbf{W}^{[\bullet]_t}$ at some node $[\bullet]_t$ cannot be partitioned as shown above. An artificial bounding constraint on the discretionary (aggregate level) variables can then be added to the problem at that node thereby giving $\mathbf{W}^{[\bullet]_t}$ the desired partitioning scheme, e.g.,

1. if $\mathbf{x}^{(1)[\bullet]_t}$ are bounded, say $\mathbf{x}^{(1)[\bullet]_t} \leq \mathbf{x}^{\text{up}}$ where all elements of $\mathbf{x}^{\text{up}}$ are finite, then add the constraint $\mathbf{1}'\mathbf{x}^{(1)[\bullet]_t} \leq \mathbf{1}'\mathbf{x}^{\text{up}}$, otherwise
2. $\mathbf{x}^{(1)[\bullet]_t}$ are unbounded, add a constraint similar to that in item 1 except replace $\mathbf{1}'\mathbf{x}^{\text{up}}$ with some arbitrarily large number, say M.

If any constraint of the type in item 2 above is tight when the solution is obtained, then the procedure should be repeated with continually increasing values for M until a solution is obtained such that all such constraints have slack.

Birge and Louveaux [17, page 132] indicate that block-separable recourse should be exploited in computational procedures whenever possible since "it may reduce work by orders of magnitude". Results presented in Chapter 6 support this claim. It is therefore surprising that very few, if any, computational studies on block-separable recourse have been performed to date – none were found in an extensive literature search.

Chapter 3
Decomposition of Linear Programs

Mathematical decomposition is usually the most efficient method, and often the only reasonable method, for obtaining solutions to large scale linear programs. Two of the most widely applied procedures, L-Shaped and Dantzig-Wolfe decomposition, are the focus of this chapter. Detailed derivations and application algorithms for these two methods are given in the first two sections below. Decomposition implementation issues are discussed in the third and final section of this chapter.

3.1 L-Shaped Decomposition

Van Slyke and Wets [196] developed L-Shaped decomposition (LSD) to solve linear programs with the following structure:

$$
\begin{array}{rlrcrcll}
z^* = & \max & \mathbf{c}'^{(0)}\mathbf{x}^{(0)} & + & \mathbf{c}'^{(1)}\mathbf{x}^{(1)} & & & \text{(3.1a)} \\
& \text{s.t.} & \mathbf{A}^{(0,0)}\mathbf{x}^{(0)} & & & \leq & \mathbf{b}^{(0)}, & \text{(3.1b)} \\
& & \mathbf{A}^{(1,0)}\mathbf{x}^{(0)} & + & \mathbf{A}^{(1,1)}\mathbf{x}^{(1)} & \leq & \mathbf{b}^{(1)}, & \text{(3.1c)} \\
& & & & \mathbf{x}^{(j)} & \geq & \mathbf{0}, \quad j = 0, 1, & \text{(3.1d)}
\end{array}
\qquad [3.1]
$$

where $\mathbf{x}^{(0)}$ and $\mathbf{x}^{(1)}$ are termed linking and linked variables respectively. The LSD algorithm is an iterative two-phase procedure that determines if problem [3.1] is infeasible or unbounded; otherwise, it obtains either an exact solution to problem [3.1] or an ϵ-optimal solution with lower and upper bounds on z^* to within a specified tolerance ϵ. The algorithm derives its name from the shape formed by the left-hand-sides of constraints (3.1b-c). Phase one of the algorithm obtains at each iteration G a solution to the LSD relaxed master

problem (RMP), **LSD RMP**:

$$
\begin{array}{rlrcll}
z_{\mathrm{UB}}^{(G)} = \max & \mathbf{c}'^{(0)}\mathbf{x}^{(0)} & + \theta & & & (3.2a)\\
\text{s.t.} & \mathbf{A}^{(0,0)}\mathbf{x}^{(0)} & & \leq & \mathbf{b}^{(0)}, & (3.2b)\\
& \boldsymbol{\pi}_k'\mathbf{A}^{(1,0)}\mathbf{x}^{(0)} & + \theta & \leq & \mathbf{b}'^{(1)}\boldsymbol{\pi}_k, \quad k = 0,\ldots,K_{\mathrm{o}}^{(G)}, & (3.2c)\\
& \boldsymbol{\delta}_k'\mathbf{A}^{(1,0)}\mathbf{x}^{(0)} & & \leq & \mathbf{b}'^{(1)}\boldsymbol{\delta}_k, \quad k = 0,\ldots,K_{\mathrm{f}}^{(G)}, & (3.2d)\\
& & \mathbf{x}^{(0)} & \geq & \mathbf{0}, & (3.2e)\\
& & \theta & \text{free}, & & (3.2f)
\end{array}
\qquad [3.2]
$$

where constraint sets (3.2c) and (3.2d) are termed *optimality* cuts and *feasibility* cuts respectively. Variable θ is referred to as a *relaxation* variable. Problem [3.1] is infeasible if problem [3.2] is infeasible, otherwise problem [3.2] is feasible and phase two of the algorithm obtains a solution to the LSD subproblem (SUB), **LSD SUB**:

$$
\begin{array}{rlcl}
\mathfrak{L}(\mathbf{x}^{(0)}) = \max & \mathbf{c}'^{(1)}\mathbf{x}^{(1)} & & \\
\text{s.t.} & \mathbf{A}^{(1,1)}\mathbf{x}^{(1)} & \leq & \mathbf{b}^{(1)} - \mathbf{A}^{(1,0)}\mathbf{x}^{(0)},\\
& \mathbf{x}^{(1)} & \geq & \mathbf{0},
\end{array}
\qquad [3.3]
$$

where $\mathbf{x}^{(0)}$ is replaced with the optimal solution to the **LSD RMP**, say $\mathbf{x}^{(0,G)}$. Function $\mathfrak{L}(\mathbf{x}^{(0)})$ defined by problem [3.3] is known as the *relaxation* function. The results of **LSD RMP** and **LSD SUB** are analyzed during each iteration to determine if an ϵ-optimal solution to problem [3.1] has been obtained at the conclusion of the second phase. The algorithm is terminated if an ϵ-optimal solution is available, otherwise either an optimality cut or a feasibility cut is added to **LSD RMP** based upon the solution to **LSD SUB** and the next iteration is executed. Details of the procedure are presented in the subsections below.

Van Slyke and Wets [196] give a detailed derivation for the LSD algorithm for the case of a single set of linked variables ($\mathbf{x}^{(1)}$ above). That version of the algorithm is referred to as the **LSD(single-cut)** algorithm since a single constraint, either an optimality cut or a feasibility cut, is added to the **LSD RMP** at each iteration. Birge and Louveaux [16] and

[17, Section 5.3] present a multicut version for the case of multiple sets of linked variables when the problem is known to be bounded. A multicut version, **LSD(multicut)**, is derived below with no restriction on the finiteness of the problem.

3.1.1 LSD for Multiple Sets of Linked Variables

Algorithm **LSD(multicut)** is developed for linear programs with multiple sets of linked variables. These linear programs have the special structure illustrated by problem **LBALP**:

$$
\begin{array}{rlrcll}
z^* = & \max & \mathbf{c}'^{(0)}\mathbf{x}^{(0)} + \sum_{j=1}^{J} \mathbf{c}'^{(j)}\mathbf{x}^{(j)} & & & \text{(3.4a)} \\
& \text{s.t.} & \mathbf{A}^{(0,0)}\mathbf{x}^{(0)} & \leq & \mathbf{b}^{(0)}, & \text{(3.4b)} \\
& & \mathbf{A}^{(j,0)}\mathbf{x}^{(0)} + \mathbf{A}^{(j,j)}\mathbf{x}^{(j)} & \leq & \mathbf{b}^{(j)}, \quad j = 1, \ldots, J, & \text{(3.4c)} \\
& & \mathbf{x}^{(j)} & \geq & \mathbf{0}, \quad j = 0, \ldots, J. & \text{(3.4d)}
\end{array}
\qquad [3.4]
$$

The matrices of coefficients in constraints (3.4b-c) form the lower block-angular structure depicted in Figure 3.1.

Problem [3.4] can be represented by an equivalent master-subproblem formulation, **LSD Master-Sub**, consisting of the phase one master problem, **LSD Master**:

$$
\begin{array}{rlrcll}
z^* = & \max & \mathbf{c}'^{(0)}\mathbf{x}^{(0)} + \sum_{j=1}^{J} \theta_j & & & \text{(3.5a)} \\
& \text{s.t.} & \mathbf{A}^{(0,0)}\mathbf{x}^{(0)} & \leq & \mathbf{b}^{(0)}, & \text{(3.5b)} \\
& & -\mathfrak{L}_j^{\mathrm{P}}(\mathbf{x}^{(0)}) + \theta_j & \leq & 0, \quad j = 1, \ldots, J, & \text{(3.5c)} \\
& & \mathbf{x}^{(0)} & \geq & \mathbf{0}, & \text{(3.5d)} \\
& & \theta_j & \text{free}, & \quad j = 1, \ldots, J, & \text{(3.5e)}
\end{array}
\qquad [3.5]
$$

and the phase two subproblems, **LSD P-SUB**(j), $j = 1, \ldots, J$:

$$
\begin{array}{rlrcl}
\mathfrak{L}_j^{\mathrm{P}}(\mathbf{x}^{(0)}) = & \max & \mathbf{c}'(j)\mathbf{x}^{(j)} & & \\
& \text{s.t.} & \mathbf{A}^{(j,j)}\mathbf{x}^{(j)} & \leq & \mathbf{b}^{(j)} - \mathbf{A}^{(j,0)}\mathbf{x}^{(0)}, \\
& & \mathbf{x}^{(j)} & \geq & \mathbf{0}.
\end{array}
\qquad [3.6]
$$

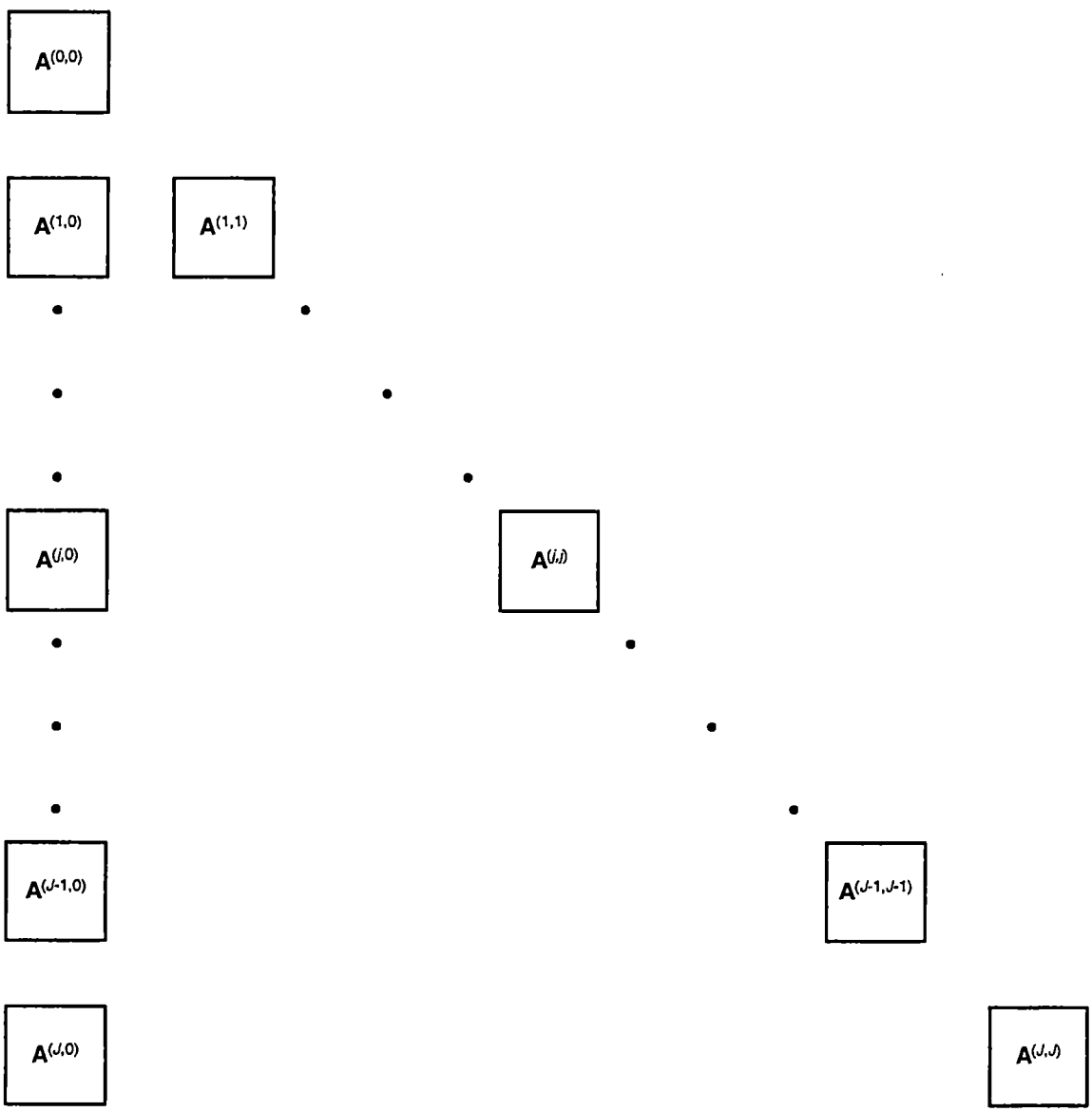


Figure 3.1: Lower Block-Angular Structure

The **LSD(multicut)** algorithm is based upon iteratively solving a relaxed formulation, **LSD Relaxed**, of the **LSD Master-Sub** formulation. Formulation **LSD Relaxed** replaces constraint sets (3.5c) with sets of optimality and feasibility cuts. The dual problems, **LSD D-SUB**(j), $j = 1, \ldots, J$:

$$\begin{array}{rlrll} \mathfrak{L}_j^{\mathrm{D}}(\mathbf{x}^{(0)}) = & \min & \left(\mathbf{b}^{(j)} - \mathbf{A}^{(j,0)}\mathbf{x}^{(0)}\right)' \boldsymbol{\pi}^{(j)} & & \text{(3.7a)} \\ & \text{s.t.} & \mathbf{A}'^{(j,j)}\boldsymbol{\pi}^{(j)} & \geq \mathbf{c}^{(j)}, & \text{(3.7b)} \\ & & \boldsymbol{\pi}^{(j)} & \geq \mathbf{0}, & \text{(3.7c)} \end{array} \qquad [3.7]$$

to **LSD P-SUB**(j), $j = 1, \ldots, J$, will be referenced frequently during algorithm development. Properties of the **LSD P-SUB**(j) relaxation function $\mathfrak{L}_j^{\mathrm{P}}\left(\mathbf{x}^{(0)}\right)$ for $j \in \mathsf{J}$ for indexing set $\mathsf{J} = \{1, \ldots, J\}$ are described in the next subsection. The **LSD Relaxed** formulation, **LSD(multicut)** algorithm development, and algorithm properties are described in subsequent subsections.

3.1.2 Properties of the LSD Relaxation Function

Define the feasibility sets, $\mathsf{X}^{(j)}\left(\mathbf{x}^{(0)}\right)$, for the linked variables as:

$$\mathsf{X}^{(j)}\left(\mathbf{x}^{(0)}\right) = \left\{\mathbf{x}^{(j)} \,\middle|\, \mathbf{A}^{(j,j)}\mathbf{x}^{(j)} \leq \mathbf{b}^{(j)} - \mathbf{A}^{(j,0)}\mathbf{x}^{(0)},\ \mathbf{x}^{(j)} \geq \mathbf{0}\right\},\ j = 1, \ldots, J,$$

where, by definition, $\mathfrak{L}_j^{\mathrm{P}}\left(\mathbf{x}^{(0)}\right)$ assumes the value of negative infinity if $\mathsf{X}^{(j)}\left(\mathbf{x}^{(0)}\right)$ is empty for a given $\mathbf{x}^{(0)}$, i.e., $\mathfrak{L}_j^{\mathrm{P}}\left(\mathbf{x}^{(0)}\right) = -\infty$ if $\mathsf{X}^{(j)}\left(\mathbf{x}^{(0)}\right) = \emptyset$. Define the feasibility sets, $\mathsf{X}_j^{(0)}$, for the linking variables as:

$$\mathsf{X}_j^{(0)} = \begin{cases} \left\{\mathbf{x}^{(0)} \,\middle|\, \mathfrak{L}_j^{\mathrm{P}}\left(\mathbf{x}^{(0)}\right) > -\infty\right\}, & \text{if } j \in \mathsf{J}, \\ \left\{\mathbf{x}^{(0)} \,\middle|\, \mathbf{A}^{(0,0)}\mathbf{x}^{(0)} \leq \mathbf{b}^{(0)},\ \mathbf{x}^{(0)} \geq \mathbf{0}\right\}, & \text{if } j = 0. \end{cases}$$

The following three propositions establish important properties for the **LSD P-SUB**(j) functions, $\mathfrak{L}_j^{\mathrm{P}}\left(\mathbf{x}^{(0)}\right), j = 1, \ldots, J$, and they are analogous to Propositions 2-4 of Van Slyke and Wets [196, pages 644-645].

Proposition 1 *For $j \in \mathsf{J}$ and for all $\mathbf{x}^{(0)} \in \mathsf{X}_j^{(0)}$, $\mathfrak{L}_j^{\mathrm{P}}\left(\mathbf{x}^{(0)}\right)$ is either a finite concave function or $\mathfrak{L}_j^{\mathrm{P}}\left(\mathbf{x}^{(0)}\right)$ is identically positive infinity.*

Proof Since problem **LSD P-SUB**(j) [3.6] is feasible for all $\mathbf{x}^{(0)} \in \mathsf{X}_j^{(0)}$, it follows from LP duality theory that problem **LSD P-SUB**(j) is unbounded if and only if the dual problem **LSD D-SUB**(j) [3.7] is infeasible. Therefore, $\mathfrak{L}_j^{\mathrm{P}}\left(\mathbf{x}^{(0)}\right)$ is either finite or identically positive infinity for all $\mathbf{x}^{(0)} \in \mathsf{X}_j^{(0)}$since problem **LSD D-SUB**(j) is infeasible if and only if constraints (3.7b), $\mathbf{A}'^{(j,j)}\boldsymbol{\pi}^{(j)} \geq \mathbf{c}^{(j)}$, are inconsistent. Assume $\mathfrak{L}_j^{\mathrm{P}}\left(\mathbf{x}^{(0)}\right)$ is finite and let $\mathbf{x}_1^{(0)}, \mathbf{x}_2^{(0)} \in \mathsf{X}_j^{(0)}$. Then $\mathbf{x}_\lambda^{(0)} = \lambda\mathbf{x}_1^{(0)} + (1-\lambda)\,\mathbf{x}_2^{(0)} \in \mathsf{X}_j^{(0)}$, for $0 \leq \lambda \leq 1$, since $\mathsf{X}_j^{(0)}$ is a convex set. Let $\mathbf{x}_1^{(j)}$, $\mathbf{x}_2^{(j)}$, and $\mathbf{x}_\lambda^{(j)}$ be optimal solutions to problem **LSD P-SUB**(j) when $\mathbf{x}^{(0)}$ equals $\mathbf{x}_1^{(0)}$, $\mathbf{x}_2^{(0)}$, and $\mathbf{x}_\lambda^{(0)}$ respectively. Then

$$\begin{aligned}
\lambda\mathfrak{L}_j^{\mathrm{P}}\left(\mathbf{x}_1^{(0)}\right) + (1-\lambda)\,\mathfrak{L}_j^{\mathrm{P}}\left(\mathbf{x}_2^{(0)}\right) &= \lambda\mathbf{c}'^{(j)}\mathbf{x}_1^{(j)} + (1-\lambda)\,\mathbf{c}'^{(j)}\mathbf{x}_2^{(j)} \\
&= \mathbf{c}'^{(j)}\left[\lambda\mathbf{x}_1^{(j)} + (1-\lambda)\,\mathbf{x}_2^{(j)}\right] \\
&\leq \mathbf{c}'^{(j)}\mathbf{x}_\lambda^{(j)} = \mathfrak{L}_j^{\mathrm{P}}\left(\mathbf{x}_\lambda^{(0)}\right).
\end{aligned}$$

Where the inequality above is justified by $[\lambda\mathbf{x}_1^{(j)} + (1-\lambda)\,\mathbf{x}_2^{(j)}] \in \mathsf{X}^{(j)}\left(\mathbf{x}_\lambda^{(0)}\right)$, i.e., feasible to **LSD P-SUB**(j) since $\mathsf{X}^{(j)}\left(\mathbf{x}_\lambda^{(0)}\right)$ is a convex set, but $[\lambda\mathbf{x}_1^{(j)} + (1-\lambda)\,\mathbf{x}_2^{(j)}]$ is not necessarily an optimal solution to problem **LSD P-SUB**(j). ■

Proposition 2 *For $j \in \mathsf{J}$, let $\widetilde{\mathbf{x}}^{(0)} \in \mathsf{X}_j^{(0)}$ and assume $\mathfrak{L}_j^{\mathrm{P}}(\widetilde{\mathbf{x}}^{(0)})$ is finite and let $\widetilde{\boldsymbol{\pi}}^{(j)}$ be a corresponding optimal solution to problem* **LSD D-SUB**(j) [3.7]. *Then the affine function*

$$\mathbf{b}'^{(j)}\widetilde{\boldsymbol{\pi}}^{(j)} - \left(\widetilde{\boldsymbol{\pi}}'^{(j)}\mathbf{A}^{(j,0)}\right)\mathbf{x}^{(0)} \qquad [3.8]$$

is a support of $\mathfrak{L}_j^{\mathrm{P}}\left(\mathbf{x}^{(0)}\right)$.

Proof By strong duality, $\mathfrak{L}_j^{\mathrm{D}}\left(\widetilde{\mathbf{x}}^{(0)}\right) = \left(\mathbf{b}^{(j)} - \mathbf{A}^{(j,0)}\widetilde{\mathbf{x}}^{(0)}\right)'\widetilde{\boldsymbol{\pi}}^{(j)} = \mathfrak{L}_j^{\mathrm{P}}\left(\widetilde{\mathbf{x}}^{(0)}\right)$. By assumption and Proposition 1, $\mathfrak{L}_j^{\mathrm{P}}\left(\mathbf{x}^{(0)}\right)$ is finite for all $\mathbf{x}^{(0)} \in \mathsf{X}_j^{(0)}$. Since $\widetilde{\boldsymbol{\pi}}^{(j)}$ is feasible but not necessarily optimal to problem **LSD D-SUB**(j) for all $\mathbf{x}^{(0)} \in \mathsf{X}_j^{(0)}$, weak duality guarantees that

$$\left(\mathbf{b}^{(j)} - \mathbf{A}^{(j,0)}\mathbf{x}^{(0)}\right)'\widetilde{\boldsymbol{\pi}}^{(j)} \geq \mathfrak{L}_j^{\mathrm{P}}\left(\mathbf{x}^{(0)}\right) \;\forall\, \mathbf{x}^{(0)} \in \mathsf{X}_j^{(0)}. \blacksquare$$

Proposition 3 *For $j \in \mathsf{J}$, assume $\mathfrak{L}_j^{\mathrm{P}}\left(\mathbf{x}^{(0)}\right)$ is finite on $\mathsf{X}_j^{(0)}$. Then $\mathfrak{L}_j^{\mathrm{P}}\left(\mathbf{x}^{(0)}\right)$ is a concave polyhedral function.*

Proof There are only a finite number of supports of type [3.8] since there are only a finite number of $\boldsymbol{\pi}^{(j)}$ that are solutions to problem **LSD D-SUB**(j) [3.7] since each such $\boldsymbol{\pi}^{(j)}$ corresponds to a basis of $\hat{\mathbf{A}}'^{(j,j)} = \left[\mathbf{A}'^{(j,j)}, \mathbf{-I}\right]$, where **-I** is the negative identity matrix of coefficients for a vector of slack variables added to constraints (3.7b), and $\hat{\mathbf{A}}'^{(j,j)}$ has only a finite number of square non-singular submatrices. By Proposition 2, there is a support of type [3.8] for all $\mathbf{x}^{(0)} \in \mathsf{X}_j^{(0)}$. Furthermore, there is a support of type [3.8] that meets

$\mathfrak{L}_j^{\mathrm{P}}\left(\mathbf{x}^{(0)}\right)$ at $\mathbf{x}^{(0)}$ for all $\mathbf{x}^{(0)} \in \mathsf{X}_j^{(0)}$. Therefore, the lower envelope of this finite number of affine supports coincides with $\mathfrak{L}_j^{\mathrm{P}}\left(\mathbf{x}^{(0)}\right)$. ■

3.1.3 LSD Relaxed Formulation

The iterative **LSD(multicut)** algorithm is based upon solving a relaxed formulation of the **LSD Master-Sub** formulation, problems [3.5] and [3.6]. Formulation **LSD Relaxed** consists of the phase one relaxed master problem, **LSD RMP:**

$$
\begin{array}{rlrcll}
\tilde{z}_{\mathrm{UB}}^{\mathrm{P}} = & \max & \mathbf{c}'^{(0)}\mathbf{x}^{(0)} + \sum_{j=1}^{J} u_j\theta_j & & & \text{(3.9a)} \\
& \text{s.t.} & \mathbf{A}^{(0,0)}\mathbf{x}^{(0)} & \leq & \mathbf{b}^{(0)}, & \text{(3.9b)} \\
& & (\mathbf{x}^{(0)}, \theta_j) & \in & \mathsf{C}_j, \quad j = 1, \ldots, J, & \text{(3.9c)} \\
& & \mathbf{x}^{(0)} & \geq & \mathbf{0}, & \text{(3.9d)} \\
& & \theta_j & \text{free}, & j = 1, \ldots, J, & \text{(3.9e)}
\end{array}
\qquad [3.9]
$$

and the phase two subproblems **LSD P-SUB**(j), $j = 1, \ldots, J$, problems [3.6]. The sets of cuts, C_j, $j = 1, \ldots, J$, in constraints (3.9c) contain optimality and feasibility cuts defined such that if $\mathbf{x}^{(j)}$, $j = 0, \ldots, J$, are feasible to the **LSD Master-Sub** formulation and thus to the original problem **LBALP** [3.4], then there exist θ_j, $j = 1, \ldots, J$, such that $\mathbf{x}^{(j)}$ and θ_j are feasible to the **LSD Relaxed** formulation with $z^* \leq z_{\mathrm{UB}} = \mathbf{c}'^{(0)}\mathbf{x}^{(0)} + \sum_{j=1}^{J} \theta_j$. Each C_j, $j \in \mathsf{J}$, is initially empty and may be augmented with an additional cut during each iteration of the algorithm. Coefficient u_j, $j \in \mathsf{J}$, in objective function (3.9a) is 0 if the corresponding cut set C_j contains no optimality cuts or 1 otherwise.

Let $\mathcal{Z}_{\mathrm{UB}}\left(\mathbf{x}^{(0)}\right) = \mathbf{c}'^{(0)}\mathbf{x}^{(0)}$ represent the contribution of $\mathbf{x}^{(0)}$ to the **LSD RMP** objective function value. Then each primary type of cut, optimality and feasibility, consists of two subtypes depending upon the finiteness of $\mathcal{Z}_{\mathrm{UB}}\left(\mathbf{x}^{(0)}\right)$. The four types of cuts are

termed feasibility (unbounded), optimality (unbounded), feasibility (finite), and optimality (finite). The applicability and derivation of each cut is discussed below.

1. $\mathcal{Z}_{\text{UB}}\left(\mathbf{x}^{(0)}\right) = \infty$: the simplex algorithm returns an extreme point, $\mathbf{x}_{\text{p}}^{(0)}$, and direction, $\mathbf{x}_{\text{d}}^{(0)}$, which determine a ray, $\mathbf{x}_{\lambda}^{(0)} = \mathbf{x}_{\text{p}}^{(0)} + \lambda\mathbf{x}_{\text{d}}^{(0)}$, $\lambda \geq 0$, along which $\mathcal{Z}_{\text{UB}}\left(\mathbf{x}_{\lambda}^{(0)}\right)$ increases monotonically with λ. (Note that one of the finite subtypes of cuts discussed in item 2 below is made if **LSD RMP** is unbounded due solely to $\theta_j = \infty$ for one or more $j \in$ J.) Then for some $j \in J$ and with $\mathbf{x}^{(0)} = \mathbf{x}_{\text{p}}^{(0)}$, solve the modified phase two subproblem **LSD PM-SUB**(j):

$$\begin{array}{rlrcl} \mathfrak{L}_j^{\text{P}\lambda}(\mathbf{x}^{(0)}) = & \max & \mathbf{c}'^{(j)}\mathbf{x}^{(j)} + (\mathbf{c}'^{(0)}\mathbf{x}_{\text{d}}^{(0)})\lambda & & \\ & \text{s.t.} & \mathbf{A}^{(j,j)}\mathbf{x}^{(j)} + (\mathbf{A}^{(j,0)}\mathbf{x}_{\text{d}}^{(0)})\lambda & \leq & \mathbf{b}^{(j)} - \mathbf{A}^{(j,0)}\mathbf{x}^{(0)}, \\ & & \mathbf{x}^{(j)} & \geq & \mathbf{0}, \\ & & \lambda & \geq & 0, \end{array} \qquad [3.10]$$

which corresponds to the modified dual phase two subproblem **LSD DM-SUB**(j):

$$\begin{array}{rlrcll} \mathfrak{L}_j^{\text{D}\lambda}(\mathbf{x}^{(0)}) = & \min & \left(\mathbf{b}^{(j)} - \mathbf{A}^{(j,0)}\mathbf{x}^{(0)}\right)'\boldsymbol{\pi}^{(j)} & & & (3.11a) \\ & \text{s.t.} & \mathbf{A}'^{(j,j)}\boldsymbol{\pi}^{(j)} & \geq & \mathbf{c}^{(j)}, & (3.11b) \\ & & (\mathbf{A}^{(j,0)}\mathbf{x}_{\text{d}}^{(0)})'\boldsymbol{\pi}^{(j)} & \geq & \mathbf{c}'^{(0)}\mathbf{x}_{\text{d}}^{(0)}, & (3.11c) \\ & & \boldsymbol{\pi}^{(j)} & \geq & \mathbf{0}. & (3.11d) \end{array} \qquad [3.11]$$

Note that if $\mathbf{x}^{(j)}$ is feasible to problem **LSD P-SUB**(j) [3.6] for some $\mathbf{x}^{(0)}$, then it is feasible to **LSD PM-SUB**(j) and $\mathfrak{L}_j^{\text{P}}(\mathbf{x}^{(0)}) \leq \mathfrak{L}_j^{\text{P}\lambda}(\mathbf{x}^{(0)})$ since $\mathbf{x}_{\text{d}}^{(0)}$ is an ascent direction for **LSD RMP** [3.9] implying that $(\mathbf{c}'^{(0)}\mathbf{x}_{\text{d}}^{(0)})\lambda \geq 0$ for $\lambda \geq 0$. The type of cut to be added to C_j depends upon the solution to problem **LSD PM-SUB**(j) as follows:

(a) **LSD PM-SUB**(j) is infeasible $\Longrightarrow$ feasibility (unbounded) cut: with $\mathbf{x}^{(0)} = \mathbf{x}_{\text{p}}^{(0)}$, solve the following simplex phase one type problem based upon problem **LSD**

PM-SUB(j):

$$\begin{array}{rlrrrrrl} \mathcal{W}_j^{\lambda}(\mathbf{x}^{(0)}) = & \min & \mathbf{1}'\mathbf{v} & & & & & \\ & \text{s.t.} & \mathbf{I}\mathbf{v} & - & \mathbf{A}^{(j,j)}\mathbf{x}^{(j)} & - & (\mathbf{A}^{(j,0)}\mathbf{x}_{\mathrm{d}}^{(0)})\lambda & \geq \mathbf{A}^{(j,0)}\mathbf{x}^{(0)} - \mathbf{b}^{(j)}, \\ & & & & & & \mathbf{x}^{(j)} & \geq \mathbf{0}, \\ & & & & & & \lambda & \geq 0, \\ & & & & & & \mathbf{v} & \geq \mathbf{0}, \end{array} \qquad [3.12]$$

which is always feasible and bounded and corresponds to the dual problem:

$$\begin{array}{rlrcll} \mathcal{W}_j^{\lambda}(\mathbf{x}^{(0)}) = & \max & \left(\mathbf{A}^{(j,0)}\mathbf{x}^{(0)} - \mathbf{b}^{(j)}\right)' \boldsymbol{\delta}^{(j)} & & & (3.13a) \\ & \text{s.t.} & \mathbf{A}'^{(j,j)}\boldsymbol{\delta}^{(j)} & \geq & \mathbf{0}, & (3.13b) \\ & & (\mathbf{A}^{(j,0)}\mathbf{x}_{\mathrm{d}}^{(0)})'\boldsymbol{\delta}^{(j)} & \geq & 0, & (3.13c) \\ & & \boldsymbol{\delta}^{(j)} & \leq & \mathbf{1}, & (3.13d) \\ & & \boldsymbol{\delta}^{(j)} & \geq & \mathbf{0}. & \end{array} \qquad [3.13]$$

Let the optimal solutions be $\mathbf{v}_{\mathrm{U}}$, $\mathbf{x}_{\mathrm{U}}^{(j)}$, λ_{U}, and $\boldsymbol{\delta}_{\mathrm{U}}^{(j)}$, and note that

$$\mathcal{W}_j^{\lambda}\left(\mathbf{x}_{\mathrm{p}}^{(0)}\right) = \mathbf{1}'\mathbf{v}_{\mathrm{U}} = (\mathbf{A}^{(j,0)}\mathbf{x}_{\mathrm{p}}^{(0)} - \mathbf{b}^{(j)})'\boldsymbol{\delta}_{\mathrm{U}}^{(j)} > 0$$

since problem **LSD PM-SUB(j)** [3.10] is infeasible. The above inequality and constraints (3.13c) imply that for $\lambda \geq 0$,

$$\begin{array}{rcl} (\mathbf{A}^{(j,0)}\mathbf{x}_{\mathrm{p}}^{(0)} - \mathbf{b}^{(j)})'\boldsymbol{\delta}_{\mathrm{U}}^{(j)} + \lambda(\mathbf{A}^{(j,0)}\mathbf{x}_{\mathrm{d}}^{(0)})'\boldsymbol{\delta}_{\mathrm{U}}^{(j)} & = & (\mathbf{A}^{(j,0)}\mathbf{x}_{\mathrm{p}}^{(0)} + \lambda\mathbf{A}^{(j,0)}\mathbf{x}_{\mathrm{d}}^{(0)} - \mathbf{b}^{(j)})'\boldsymbol{\delta}_{\mathrm{U}}^{(j)} \\ & = & (\mathbf{A}^{(j,0)}\mathbf{x}_{\lambda}^{(0)} - \mathbf{b}^{(j)})'\boldsymbol{\delta}_{\mathrm{U}}^{(j)} \\ & > & 0, \end{array}$$

so that the feasibility (unbounded) cut

$$(\boldsymbol{\delta}_{\mathrm{U}}'^{(j)}\mathbf{A}^{(j,0)})\mathbf{x}^{(0)} \leq \mathbf{b}'^{(j)}\boldsymbol{\delta}_{\mathrm{U}}^{(j)} \qquad [3.14]$$

will not admit the ray $\mathbf{x}_\lambda^{(0)} = \mathbf{x}_\mathrm{p}^{(0)} + \lambda \mathbf{x}_\mathrm{d}^{(0)}$, $\lambda \geq 0$, but will not exclude any $\mathbf{x}^{(0)} \in \mathsf{X}_j^{(0)}$ since $\mathcal{W}_j^\lambda\left(\mathbf{x}^{(0)}\right) = 0 \geq (\mathbf{A}^{(j,0)}\mathbf{x}^{(0)} - \mathbf{b}^{(j)})'\boldsymbol{\delta}_\mathrm{U}^{(j)}$ because $\boldsymbol{\delta}_\mathrm{U}^{(j)}$ is feasible but not necessarily optimum to problem [3.13] for all $\mathbf{x}^{(0)} \in \mathsf{X}_j^{(0)}$.

(b) **LSD PM-SUB**(j) is feasible and bounded $\Longrightarrow$ optimality (unbounded) cut: let the optimal solutions to problems **LSD PM-SUB**(j) [3.10] and **LSD DM-SUB**(j) [3.11] be $\mathbf{x}_\mathrm{U}^{(j)}$, λ_U, and $\boldsymbol{\pi}_\mathrm{U}^{(j)}$. Strong duality implies that $\mathfrak{L}_j^{\mathrm{P}\lambda}(\mathbf{x}_\mathrm{p}^{(0)}) = \mathfrak{L}_j^{\mathrm{D}\lambda}(\mathbf{x}_\mathrm{p}^{(0)}) = (\mathbf{b}^{(j)} - \mathbf{A}^{(j,0)}\mathbf{x}_\mathrm{p}^{(0)})'\boldsymbol{\pi}_\mathrm{U}^{(j)}$. Then, by weak duality,

$$\mathfrak{L}_j^{\mathrm{P}\lambda}(\mathbf{x}^{(0)}) \leq (\mathbf{b}^{(j)} - \mathbf{A}^{(j,0)}\mathbf{x}^{(0)})'\boldsymbol{\pi}_\mathrm{U}^{(j)}$$

for all $\mathbf{x}^{(0)} \in \mathsf{X}_j^{(0)}$ since $\boldsymbol{\pi}_\mathrm{U}^{(j)}$ is feasible but not necessarily optimal to problem **LSD DM-SUB**(j) for all $\mathbf{x}^{(0)}$. Furthermore, constraints (3.5c) of problem **LSD Master** [3.5] indicate that $\theta_j \leq \mathfrak{L}_j^{\mathrm{P}}(\mathbf{x}^{(0)})$ for any $\left(\mathbf{x}^{(0)}, \theta_j\right)$ feasible to **LSD Master** implying that

$$\theta_j \leq \left(\mathbf{b}^{(j)} - \mathbf{A}^{(j,0)}\mathbf{x}^{(0)}\right)' \boldsymbol{\pi}_\mathrm{U}^{(j)}$$

for any $\left(\mathbf{x}^{(0)}, \theta_j\right)$ feasible to **LSD Master**. Therefore, the optimality (unbounded) cut

$$(\boldsymbol{\pi}_\mathrm{U}'^{(j)}\mathbf{A}^{(j,0)})\mathbf{x}^{(0)} + \theta_j \leq \mathbf{b}'^{(j)}\boldsymbol{\pi}_\mathrm{U}^{(j)} \qquad [3.15]$$

will not admit $\mathbf{x}_\lambda^{(0)} = \mathbf{x}_\mathrm{p}^{(0)} + \lambda \mathbf{x}_\mathrm{d}^{(0)}$, $\lambda \geq 0$, for arbitrarily large λ since constraints (3.11c) imply that $\lambda(\mathbf{A}^{(j,0)}\mathbf{x}_\mathrm{d}^{(0)})'\boldsymbol{\pi}_\mathrm{U}^{(j)} \geq \lambda \mathbf{c}'^{(0)}\mathbf{x}_\mathrm{d}^{(0)}$ which increases monotonically with λ since $\mathbf{x}_\mathrm{d}^{(0)}$ is an ascent direction for problem **LSD RMP** [3.9]. Moreover, cut [3.15] will not exclude any point, say $\left(\widetilde{\mathbf{x}}^{(0)}, \widetilde{\theta}_j\right)$, that is a part of the optimal

solution to problem **LSD Master** since $\widetilde{\mathbf{x}}^{(0)} \in \mathsf{X}_j^{(0)}$ and $\widetilde{\theta}_j \leq \mathfrak{L}_j^{\mathrm{P}}(\widetilde{\mathbf{x}}^{(0)})$ implying that:

$$\widetilde{\theta}_j \leq \mathfrak{L}_j^{\mathrm{P}}(\widetilde{\mathbf{x}}^{(0)}) \leq \mathfrak{L}_j^{\mathrm{P}\lambda}(\widetilde{\mathbf{x}}^{(0)}) = \mathfrak{L}_j^{\mathrm{D}\lambda}(\widetilde{\mathbf{x}}^{(0)}) \leq (\mathbf{b}^{(j)} - \mathbf{A}^{(j,0)}\widetilde{\mathbf{x}}^{(0)})'\boldsymbol{\pi}_{\mathrm{U}}^{(j)},$$

by strong duality and $\boldsymbol{\pi}_{\mathrm{U}}^{(j)}$ feasible but not necessarily optimum to **LSD DM-SUB**(j) for all $\mathbf{x}^{(0)}$.

(c) **LSD PM-SUB**(j) is feasible and unbounded: no cuts are added to C_j.

2. $\mathcal{Z}_{\mathrm{UB}}\left(\mathbf{x}^{(0)}\right) < \infty$: let the optimal solutions to problem **LSD RMP** [3.9] be $\widetilde{\mathbf{x}}^{(0)}$ and $\widetilde{\theta}_j$, $j = 1, \ldots, J$. For some $j \in \mathsf{J}$, solve problem **LSD P-SUB**(j) [3.6] and the type of cut to add to C_j depends upon the solution as follows:

(a) **LSD P-SUB**(j) is infeasible $\Longrightarrow$ feasibility (finite) cut: with $\mathbf{x}^{(0)} = \widetilde{\mathbf{x}}^{(0)}$, solve the following simplex phase one type problem based upon problem **LSD P-SUB**(j):

$$\begin{array}{rlrcl} \mathcal{W}_j\left(\mathbf{x}^{(0)}\right) = & \min & \mathbf{1}'\mathbf{v} & & \\ & \text{s.t.} & \mathbf{I}\mathbf{v} - \mathbf{A}^{(j,j)}\mathbf{x}^{(j)} & \geq & \mathbf{A}^{(j,0)}\mathbf{x}^{(0)} - \mathbf{b}^{(j)}, \\ & & \mathbf{x}^{(j)} & \geq & \mathbf{0}, \\ & & \mathbf{v} & \geq & \mathbf{0}, \end{array} \qquad [3.16]$$

which is always feasible and bounded and corresponds to the dual problem:

$$\begin{array}{rlrcll} \mathcal{W}_j\left(\mathbf{x}^{(0)}\right) = & \max & \left(\mathbf{A}^{(j,0)}\mathbf{x}^{(0)} - \mathbf{b}^{(j)}\right)'\boldsymbol{\delta}^{(j)} & & & (3.17a) \\ & \text{s.t.} & \mathbf{A}'^{(j,j)}\boldsymbol{\delta}^{(j)} & \geq & \mathbf{0}, & (3.17b) \\ & & \boldsymbol{\delta}^{(j)} & \leq & \mathbf{1}, & (3.17c) \\ & & \boldsymbol{\delta}^{(j)} & \geq & \mathbf{0}. & (3.17d) \end{array} \qquad [3.17]$$

Let the optimal solutions be $\mathbf{v}_{\mathrm{B}}$, $\mathbf{x}_{\mathrm{B}}^{(j)}$, and $\boldsymbol{\delta}_{\mathrm{B}}^{(j)}$, and note that

$$\mathcal{W}_j\left(\widetilde{\mathbf{x}}^{(0)}\right) = \mathbf{1}'\mathbf{v}_{\mathrm{B}} = (\mathbf{A}^{(j,0)}\widetilde{\mathbf{x}}^{(0)} - \mathbf{b}^{(j)})'\boldsymbol{\delta}_{\mathrm{B}}^{(j)} > 0$$

since problem **LSD P-SUB**(j) is infeasible. Therefore, the feasibility (finite) cut

$$(\boldsymbol{\delta}_{\mathrm{B}}'^{(j)}\mathbf{A}^{(j,0)})\mathbf{x}^{(0)} \leq \mathbf{b}'^{(j)}\boldsymbol{\delta}_{\mathrm{B}}^{(j)} \qquad [3.18]$$

will not admit the point $\widetilde{\mathbf{x}}^{(0)}$ but will not exclude any $\mathbf{x}^{(0)} \in \mathsf{X}_j^{(0)}$ since $\mathcal{W}_j\left(\mathbf{x}^{(0)}\right) = 0 \geq (\mathbf{A}^{(j,0)}\mathbf{x}^{(0)} - \mathbf{b}^{(j)})'\boldsymbol{\delta}_{\mathrm{B}}^{(j)}$ because $\boldsymbol{\delta}_{\mathrm{B}}^{(j)}$ is feasible but not necessarily optimum to problem [3.17] for all $\mathbf{x}^{(0)} \in \mathsf{X}_j^{(0)}$.

(b) **LSD P-SUB**(j) is feasible and bounded with $\mathfrak{L}_j^{\mathrm{P}}(\widetilde{\mathbf{x}}^{(0)}) < \widetilde{\theta}_j \Longrightarrow$ optimality (finite) cut: let the optimal solution to the corresponding dual problem **LSD D-SUB**(j) [3.7] be $\boldsymbol{\pi}_{\mathrm{B}}^{(j)}$. Strong duality implies that $\mathfrak{L}_j^{\mathrm{P}}(\widetilde{\mathbf{x}}^{(0)}) = \mathfrak{L}_j^{\mathrm{D}}(\widetilde{\mathbf{x}}^{(0)}) = \left(\mathbf{b}^{(j)} - \mathbf{A}^{(j,0)}\widetilde{\mathbf{x}}^{(0)}\right)'\boldsymbol{\pi}_{\mathrm{B}}^{(j)}$. Therefore, by strong duality and Proposition 1,

$$\mathfrak{L}_j^{\mathrm{P}}(\mathbf{x}^{(0)}) = \mathfrak{L}_j^{\mathrm{D}}(\mathbf{x}^{(0)}) \leq \left(\mathbf{b}^{(j)} - \mathbf{A}^{(j,0)}\mathbf{x}^{(0)}\right)'\boldsymbol{\pi}_{\mathrm{B}}^{(j)}$$

for all $\mathbf{x}^{(0)} \in \mathsf{X}_j^{(0)}$ since $\boldsymbol{\pi}_{\mathrm{B}}^{(j)}$ is feasible but not necessarily optimum to problem **LSD D-SUB**(j) for all $\mathbf{x}^{(0)}$. Furthermore, constraints (3.5c) of problem **LSD Master** [3.5] indicate that $\theta_j \leq \mathfrak{L}_j^{\mathrm{P}}(\mathbf{x}^{(0)})$ for any $\left(\mathbf{x}^{(0)}, \theta_j\right)$ feasible to **LSD Master** implying that

$$\theta_j \leq \left(\mathbf{b}^{(j)} - \mathbf{A}^{(j,0)}\mathbf{x}^{(0)}\right)'\boldsymbol{\pi}_{\mathrm{B}}^{(j)}$$

for any $\left(\mathbf{x}^{(0)}, \theta_j\right)$ feasible to **LSD Master**. Therefore, the optimality (finite) cut

$$(\boldsymbol{\pi}_{\mathrm{B}}'^{(j)}\mathbf{A}^{(j,0)})\mathbf{x}^{(0)} + \theta_j \leq \mathbf{b}'^{(j)}\boldsymbol{\pi}_{\mathrm{B}}^{(j)} \qquad [3.19]$$

will not admit the point $(\widetilde{\mathbf{x}}^{(0)}, \widetilde{\theta}_j)$ since $\widetilde{\theta}_j > \mathfrak{L}_j^{\mathrm{P}}(\widetilde{\mathbf{x}}^{(0)})$ but will not exclude any $\left(\mathbf{x}^{(0)}, \theta_j\right)$ feasible to **LSD Master**.

(c) **LSD P-SUB**(j) is feasible and bounded with $\mathfrak{L}_j^{\mathrm{P}}(\widetilde{\mathbf{x}}^{(0)}) = \widetilde{\theta}_j$ or **LSD P-SUB**(j) is feasible and unbounded: no cuts are added to C_j.

3.1.4 Algorithm LSD(multicut)

Algorithm **LSD(multicut)** is composed of two procedures – initialize and optimize. The initialization procedure adds at least one optimality (finite) type cut [3.19] to each C_j, $j = 1, \ldots, J$, if the procedure determines that the original problem **LBALP** [3.4] has a feasible and finite optimal solution. The optimization procedure finds an ϵ-optimal solution to problem **LBALP** given that the initialization procedure determined that a finite optimal solution exists. The algorithm is applied to the **LSD Relaxed** formulation, problems [3.9] and [3.6], and is detailed below followed by the descriptions for the two contained procedures. Complementary flowcharts for the algorithm and procedures follow the detailed descriptions. Note that the algorithm assumes a predefined nonnegative value for the relative difference, ϵ, between the upper and lower bounds on the solution, z^*, to problem **LBALP**.

Algorithm LSD(multicut)

Step 0: Initialize the following parameters:

$$
\begin{array}{rcll}
G & \leftarrow & 0, & \\
z_{\text{LB}} & \leftarrow & -\infty, & \\
z_{\text{UB}} & \leftarrow & \infty & \\
u_j & \leftarrow & 0, & j = 1, \ldots, J, \\
\widetilde{\theta}_j & \leftarrow & \infty, & j = 1, \ldots, J, \\
C_j & \leftarrow & \emptyset, & j = 1, \ldots, J, \\
K_o^{(j)} & \leftarrow & 0, & j = 1, \ldots, J, \\
K_f^{(j)} & \leftarrow & 0, & j = 1, \ldots, J, \\
feasible & \leftarrow & \text{true}, & \\
finite & \leftarrow & \text{true}, & \\
solved & \leftarrow & \text{false}. &
\end{array}
$$

Go to Step 1.

Step 1: Execute Procedure **LSD(multicut)-Initialize**. If $feasible$ = false: stop with an infeasible problem **LBALP** [3.4]; else if $finite$ = false: stop with unbounded problem **LBALP**; else if $solved$ = true: stop with ϵ-optimal solution to problem **LBALP**; else go to Step 2.

Step 2: Execute Procedure **LSD(multicut)-Optimize**. Stop with ϵ-optimal solution to problem **LBALP** [3.4].

Parameter G is the algorithm iteration counter and is increased by one in the two referenced procedures with each attempt to solve problem **LSD RMP** [3.9]. The best (greatest) lower and best (least) upper bounds on the value, z^*, of problem **LBALP** found through the current iteration are recorded by z_{LB} and z_{UB} respectively. For $j = 1, \ldots, J$, **LSD RMP** objective function coefficient u_j for variable θ_j is initially set to zero and $\widetilde{\theta}_j$ to positive infinity to represent the optimum value of θ_j. Coefficient u_j, $j \in J$, is set to one in procedure **LSD(multicut)-Initialize** when the first optimality cut is added to cut set C_j after which $\widetilde{\theta}_j$ may acquire a finite value. Parameters $K_{\text{o}}^{(j)}$ and $K_{\text{f}}^{(j)}$ record the number of optimality cuts and feasibility cuts respectively that are in C_j, $j = 1, \ldots, J$, at the conclusion of each iteration. Logical parameters $feasible$, $finite$, and $solved$ indicate the return status of procedure **LSD(multicut)-Initialize** as follows:

$feasible$: state false indicates that no feasible solution exists for problem **LBALP** [3.4] while state true indicates that there is a feasible solution;

finite: meaningful only when *feasible* = true and state false means problem **LBALP** is unbounded while state true indicates that a bounded solution exists;

solved: meaningful only when *feasible* = *finite* = true and state true indicates that an ϵ-optimal solution to problem **LBALP** has been determined while state false indicates that procedure **LSD(multicut)-Optimize** should be executed to obtain an ϵ-optimal solution.

Procedure LSD(multicut)-Initialize

Step 1: Set $G \leftarrow G+1$ and solve problem **LSD RMP** [3.9]. If **LSD RMP** is infeasible, set *feasible* $\leftarrow$ false and return to algorithm; else, **LSD RMP** is feasible, let optimal solution be $\widetilde{\mathbf{x}}^{(0)}$ and $\widetilde{\theta}_k$, $k = 1, \ldots, J$, (where $\widetilde{\theta}_k = \infty$ if C_k contains no optimality cuts and $\widetilde{\mathbf{x}}^{(0)}$ represents an ascent ray if $\mathcal{Z}_{\text{UB}}(\widetilde{\mathbf{x}}^{(0)}) = \infty$) and go to Step 2.

Step 2: Set $j \leftarrow 0$, *bound* $\leftarrow 0$, and *cuts* $\leftarrow 0$. If $\mathcal{Z}_{\text{UB}}(\widetilde{\mathbf{x}}^{(0)}) = \infty$ (**LSD RMP** is unbounded regardless of $\widetilde{\theta}_j$, $j = 1, \ldots, J$, values), let ascent ray be $\widetilde{\mathbf{x}}^{(0)} = \mathbf{x}_\lambda^{(0)} = \mathbf{x}_{\text{p}}^{(0)} + \lambda \mathbf{x}_{\text{d}}^{(0)}$, $\lambda \geq 0$, set $\mathfrak{L}_k^{\text{P}}(\mathbf{x}_\lambda^{(0)}) \leftarrow -\infty$, $k = 1, \ldots, J$, and go to Step 3; else, $\mathcal{Z}_{\text{UB}}(\widetilde{\mathbf{x}}^{(0)}) < \infty$, go to Step 4.

Step 3: If $j = J$, go to Step 5. Set $j \leftarrow j + 1$ and solve problem **LSD PM-SUB**(j) [3.10] with $\mathbf{x}^{(0)} = \mathbf{x}_{\text{p}}^{(0)}$. If **LSD PM-SUB**($j$) is infeasible, go to Substep 3a; else if **LSD PM-SUB**(j) is feasible and bounded, go to Substep 3b; else, **LSD PM-SUB**(j) is feasible and unbounded, repeat Step 3.

Substep 3a: Set $cuts \leftarrow cuts + 1$, solve simplex phase one type problem [3.12] with $\mathbf{x}^{(0)} = \mathbf{x}_{\mathrm{p}}^{(0)}$ and let the optimal dual multipliers be $\boldsymbol{\delta}_{\mathrm{U}}^{(j)}$. Construct feasibility (unbounded) type cut [3.14] and set

$$\mathsf{C}_j \leftarrow \mathsf{C}_j \bigcap \left\{ \mathbf{x}^{(0)} \,\middle|\, \left(\boldsymbol{\delta}_{\mathrm{U}}^{\prime(j)} \mathbf{A}^{(j,0)} \right) \mathbf{x}^{(0)} \leq \mathbf{b}^{\prime(j)} \boldsymbol{\delta}_{\mathrm{U}}^{(j)} \right\}.$$

Set $K_{\mathrm{f}}^{(j)} \leftarrow K_{\mathrm{f}}^{(j)} + 1$ and return to Step 3.

Substep 3b: Set $cuts \leftarrow cuts + 1$ and let optimal multipliers to **LSD PM-SUB**(j) be $\boldsymbol{\pi}_{\mathrm{U}}^{(j)}$. Construct optimality (unbounded) type cut [3.15] and set

$$\mathsf{C}_j \leftarrow \mathsf{C}_j \bigcap \left\{ (\mathbf{x}^{(0)}, \theta_j) \,\middle|\, \left(\boldsymbol{\pi}_{\mathrm{U}}^{\prime(j)} \mathbf{A}^{(j,0)} \right) \mathbf{x}^{(0)} + \theta_j \leq \mathbf{b}^{\prime(j)} \boldsymbol{\pi}_{\mathrm{U}}^{(j)} \right\}.$$

If $u_j = 0$, set $u_j \leftarrow 1$. Set $K_{\mathrm{o}}^{(j)} \leftarrow K_{\mathrm{o}}^{(j)} + 1$ and return to Step 3.

Step 4: If $j = J$, go to Step 5. Set $j \leftarrow j + 1$ and solve problem **LSD P-SUB**(j) [3.6] with $\mathbf{x}^{(0)} = \widetilde{\mathbf{x}}^{(0)}$. If **LSD P-SUB**($j$) is infeasible, go to Substep 4a; else if **LSD P-SUB**(j) is feasible and bounded with $\mathfrak{L}_j^{\mathrm{P}}(\widetilde{\mathbf{x}}^{(0)}) < \widetilde{\theta}_j$, go to Substep 4b; else, **LSD P-SUB**(j) is feasible and either $\mathfrak{L}_j^{\mathrm{P}}(\widetilde{\mathbf{x}}^{(0)}) = \widetilde{\theta}_j$ or $\mathfrak{L}_j^{\mathrm{P}}(\widetilde{\mathbf{x}}^{(0)}) = \infty$, go to Substep 4c.

Substep 4a: Set $cuts \leftarrow cuts + 1$ and recall that $\mathfrak{L}_j^{\mathrm{P}}(\widetilde{\mathbf{x}}^{(0)}) = -\infty$ by definition since $\mathsf{X}^{(j)}(\widetilde{\mathbf{x}}^{(0)}) = \emptyset$. Solve simplex phase one type problem [3.16] with $\mathbf{x}^{(0)} = \widetilde{\mathbf{x}}^{(0)}$and let optimal multipliers be $\boldsymbol{\delta}_{\mathrm{B}}^{(j)}$. Construct feasibility (finite) type cut [3.18] and set

$$\mathsf{C}_j \leftarrow \mathsf{C}_j \bigcap \left\{ \mathbf{x}^{(0)} \,\middle|\, \left(\boldsymbol{\delta}_{\mathrm{B}}^{\prime(j)} \mathbf{A}^{(j,0)} \right) \mathbf{x}^{(0)} \leq \mathbf{b}^{\prime(j)} \boldsymbol{\delta}_{\mathrm{B}}^{(j)} \right\}.$$

Set $K_{\mathrm{f}}^{(j)} \leftarrow K_{\mathrm{f}}^{(j)} + 1$ and return to Step 4.

Substep 4b: Let the optimal dual multipliers to **LSD P-SUB**(j) be $\boldsymbol{\pi}_{\mathrm{B}}^{(j)}$. Construct optimality (finite) type cut [3.19] and set

$$\mathsf{C}_j \leftarrow \mathsf{C}_j \bigcap \left\{ \left(\mathbf{x}^{(0)}, \theta_j\right) \middle| \left(\boldsymbol{\pi}_{\mathrm{B}}^{\prime(j)} \mathbf{A}^{(j,0)}\right) \mathbf{x}^{(0)} + \theta_j \leq \mathbf{b}^{\prime(j)} \boldsymbol{\pi}_{\mathrm{B}}^{(j)} \right\}.$$

If $u_j = 0$, set $u_j \leftarrow 1$. Set $bound \leftarrow bound + 1$. Set $K_{\mathrm{o}}^{(j)} \leftarrow K_{\mathrm{o}}^{(j)} + 1$ and return to Step 4.

Substep 4c: If $\mathfrak{L}_j^{\mathrm{P}}(\widetilde{\mathbf{x}}^{(0)}) = \widetilde{\theta}_j < \infty$, set $bound \leftarrow bound + 1$. Return to Step 4.

Step 5: Set $z_{\mathrm{UB}} \leftarrow \mathcal{Z}_{\mathrm{UB}}(\widetilde{\mathbf{x}}^{(0)}) + \sum_{j=1}^{J} \widetilde{\theta}_j$. If $cuts = 0$ and $bound < J$, set $finite \leftarrow$ false and return to algorithm; else if $bound = J$, go to Step 6; else return to Step 1.

Step 6: Set $z_{\mathrm{LB}}^{(G)} \leftarrow \mathcal{Z}_{\mathrm{UB}}(\widetilde{\mathbf{x}}^{(0)}) + \sum_{j=1}^{J} \mathfrak{L}_j^{\mathrm{P}}(\widetilde{\mathbf{x}}^{(0)})$. Set $z_{\mathrm{LB}} \leftarrow \max\left(z_{\mathrm{LB}}, z_{\mathrm{LB}}^{(G)}\right)$. If $(z_{\mathrm{UB}} - z_{\mathrm{LB}}) \leq |z_{\mathrm{LB}}|\, \epsilon$, set $solved \leftarrow$ true. Return to algorithm.

Procedure **LSD(multicut)-Initialize** determines if problem **LBALP** [3.4] is infeasible ($feasible =$ false in Step 1) or unbounded ($finite =$ false in Step 5) or, otherwise, is feasible and bounded ($feasible = finite =$ true). Two courses of action based upon the state of parameter *solved* are possible for the last case (feasible and bounded):

solved = **true:** an ϵ-optimal solution to problem **LBALP** has been determined and the algorithm should be terminated, otherwise,

solved = **false:** procedure **LSD(multicut)-Optimize** should be executed to determine an ϵ-optimal solution to problem **LBALP** – each cut set C_j, $j = 1, \ldots, J$, will contain at least one optimality (finite) type cut [3.19].

Note that Step 3 will never be executed after the first execution of Step 4. Step 4 is called, say at iteration $\widetilde{G}$, when a feasible solution to **LSD RMP** [3.9] has been found such that $\mathcal{Z}_{\text{UB}}(\mathbf{x}^{(0)}) < \infty$. Executing Step 3 after this event, $G > \widetilde{G}$, would imply a ray of unbounded increase for $\mathbf{c}'^{(0)}\mathbf{x}^{(0)}$ over a feasible region no larger than the one existing at iteration $\widetilde{G}$ which is not possible.

Procedure LSD(multicut)-Optimize

Step 1: Set $G \leftarrow G + 1$ and solve problem **LSD RMP** [3.9]. Let optimal solution be $\widetilde{\mathbf{x}}^{(0)}$ and $\widetilde{\theta}_j$, $j = 1, \ldots, J$. Set $z_{\text{UB}} \leftarrow \widetilde{z}^{\text{P}}_{\text{UB}}$, $j \leftarrow 0$ and $cuts \leftarrow 0$ and go to Step 2.

Step 2: If $j = J$, go to Step 3. Set $j \leftarrow j + 1$ and solve problem **LSD P-SUB**(j) [3.6] with $\mathbf{x}^{(0)} = \widetilde{\mathbf{x}}^{(0)}$. If **LSD P-SUB**($j$) is infeasible, go to Substep 2a; else if $\mathfrak{L}^{\text{P}}_j(\widetilde{\mathbf{x}}^{(0)}) < \widetilde{\theta}_j$, go to Substep 2b; else, **LSD P-SUB**(j) is feasible with $\mathfrak{L}^{\text{P}}_j(\widetilde{\mathbf{x}}^{(0)}) = \widetilde{\theta}_j$, repeat Step 2.

Substep 2a: Set $cuts \leftarrow cuts + 1$. Solve simplex phase 1 type problem [3.16] and let optimal multipliers be $\boldsymbol{\delta}^{(j)}_{\text{B}}$. Construct feasibility (finite) type cut [3.18] and set

$$\mathsf{C}_j \leftarrow \mathsf{C}_j \bigcap \left\{ \mathbf{x}^{(0)} \,\middle|\, \left(\boldsymbol{\delta}'^{(j)}_{\text{B}} \mathbf{A}^{(j,0)} \right) \mathbf{x}^{(0)} \leq \mathbf{b}'^{(j)} \boldsymbol{\delta}^{(j)}_{\text{B}} \right\}.$$

Set $K^{(j)}_{\text{f}} \leftarrow K^{(j)}_{\text{f}} + 1$ and return to Step 2.

Substep 2b: Let optimal multipliers to **LSD P-SUB**(j) be $\boldsymbol{\pi}^{(j)}_{\text{B}}$. Construct optimality (finite) type cut [3.19] and set

$$\mathsf{C}_j \leftarrow \mathsf{C}_j \bigcap \left\{ (\mathbf{x}^{(0)}, \theta_j) \,\middle|\, \left(\boldsymbol{\pi}'^{(j)}_{\text{B}} \mathbf{A}^{(j,0)} \right) \mathbf{x}^{(0)} + \theta_j \leq \mathbf{b}'^{(j)} \boldsymbol{\pi}^{(j)}_{\text{B}} \right\}.$$

Set $K^{(j)}_{\text{o}} \leftarrow K^{(j)}_{\text{o}} + 1$ and return to Step 2.

Step 3: If $cuts = 0$: set $z_{\text{LB}}^{(G)} \leftarrow z_{\text{UB}} - \sum_{j=1}^{J} \left[\widetilde{\theta}_j - \mathfrak{L}_j^{\text{P}}(\widetilde{\mathbf{x}}^{(0)})\right]$ and then set $z_{\text{LB}} \leftarrow \max\left(z_{\text{LB}}, z_{\text{LB}}^{(G)}\right)$. If $(z_{\text{UB}} - z_{\text{LB}}) \leq |z_{\text{LB}}|\,\epsilon$, return to algorithm; else return to Step 1.

Procedure **LSD(multicut)-Optimize** is executed only if procedure **LSD(multicut)-Initialize** determines that problem **LBALP** [3.4] is feasible and bounded. Therefore, procedure **LSD(multicut)-Optimize** obtains a bounded solution to the **LSD Relaxed** formulation, problems [3.9] and [3.6], at each iteration and must return a ϵ-optimal solution for problem **LBALP** as shown in the next subsection.

Note that the termination test, $(z_{\text{UB}} - z_{\text{LB}}) \leq |z_{\text{LB}}|\,\epsilon$, in Step 6 of procedure **LSD(multicut)-Initialize** and Step 3 of procedure **LSD(multicut)-Optimize** must be modified if there is a possibility that $z^* = 0$. Any such modification (e.g., terminate if $z_{\text{LB}} \geq -\epsilon_0$ and $z_{\text{UB}} \leq \epsilon_0$ where ϵ_0 is the smallest positive real number possible on the computer) is assumed and is not discussed further.

Flowcharts of algorithm **LSD(multicut)** and the initialization and optimization procedures are at Figures 3.2-3.4.

3.1.5 Finite Termination of Algorithm LSD(multicut)

Algorithm **LSD(multicut)** will terminate in a finite number of iterations with either an ϵ-optimal solution to problem **LBALP** [3.4] or with an indication that problem **LBALP** is infeasible or unbounded. The proof of finite termination relies upon Propositions 1-3 starting on page 50 and five additional properties of the algorithm and the underlying **LSD**

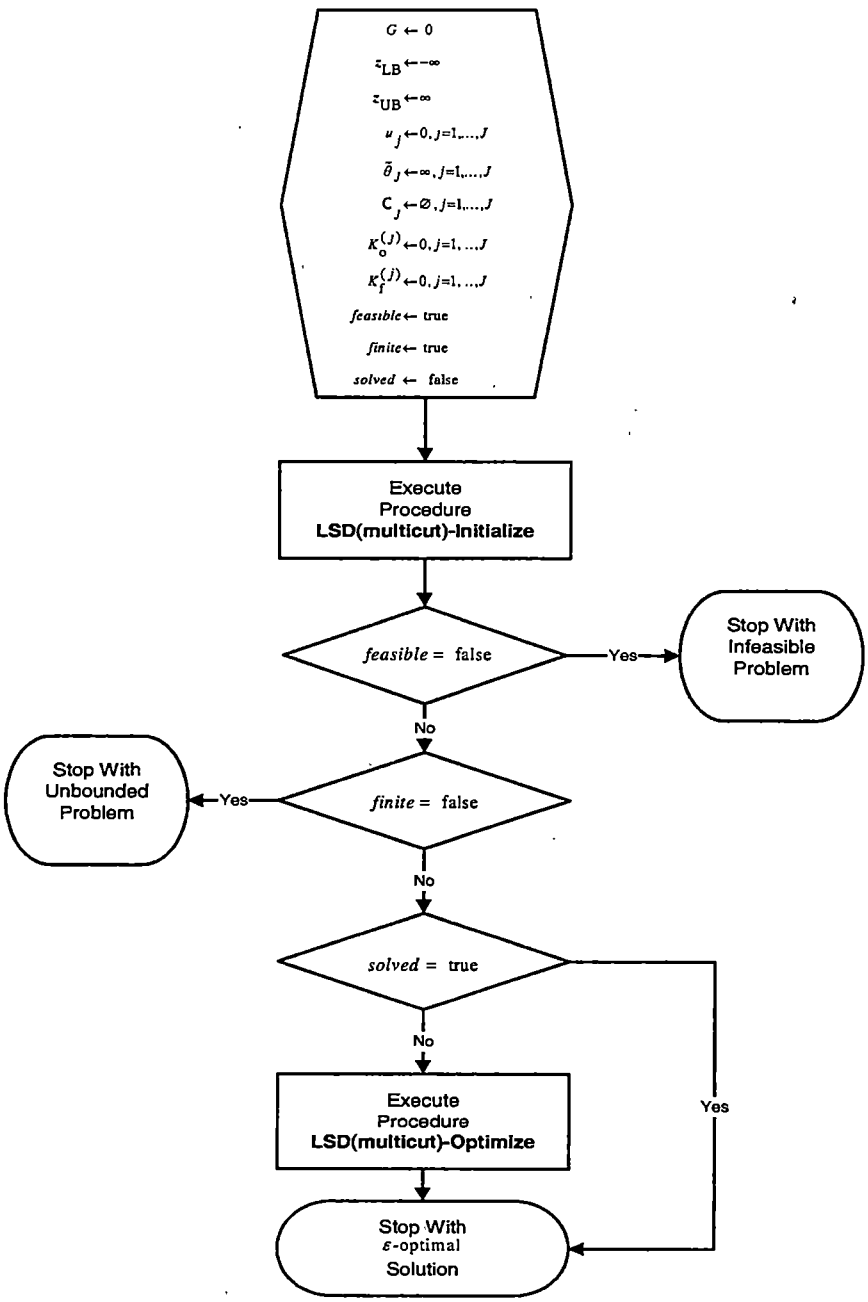


Figure 3.2: Algorithm **LSD(multicut)** Flowchart

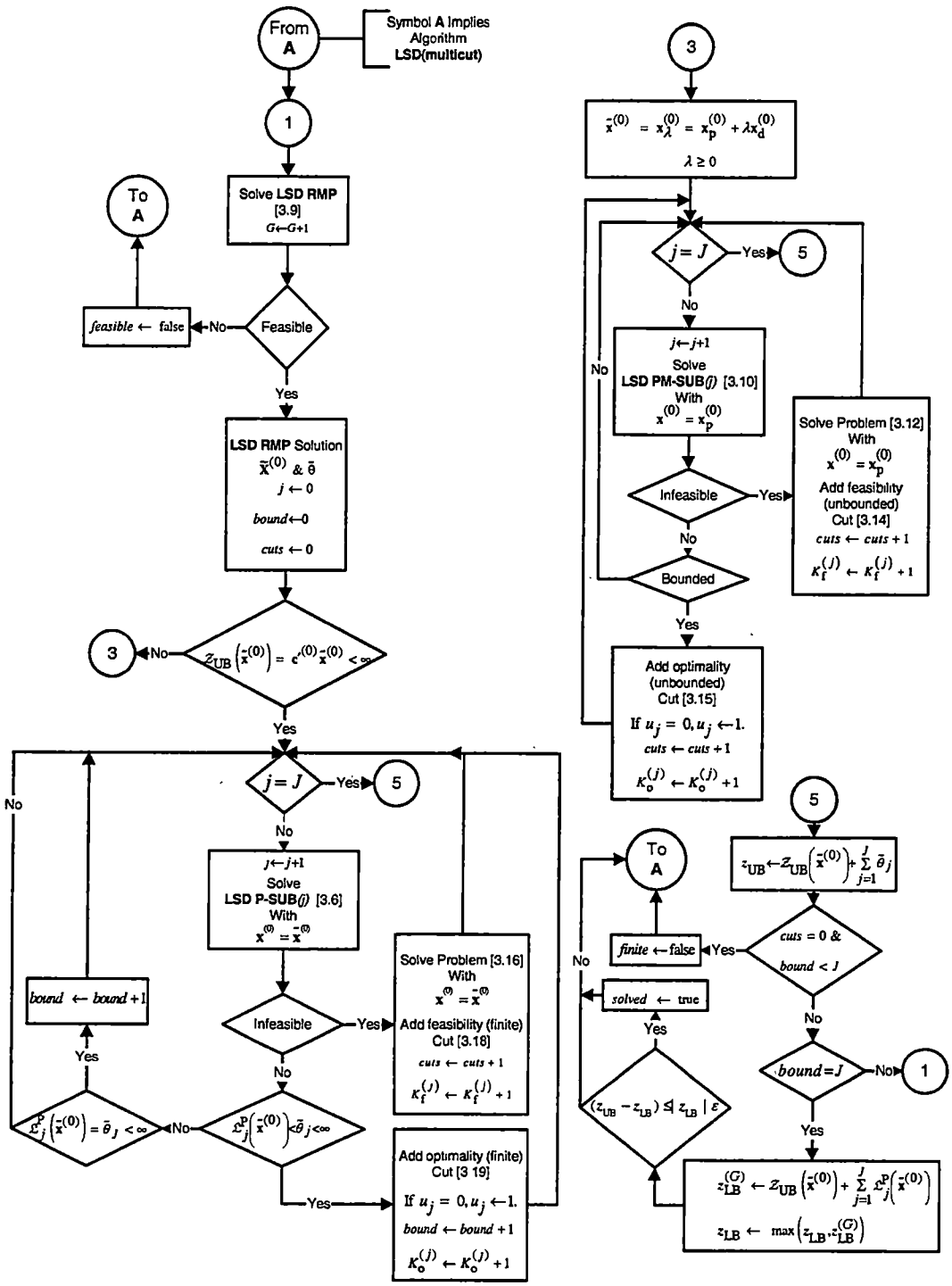


Figure 3.3: Procedure **LSD(multicut)-Initialize** Flowchart

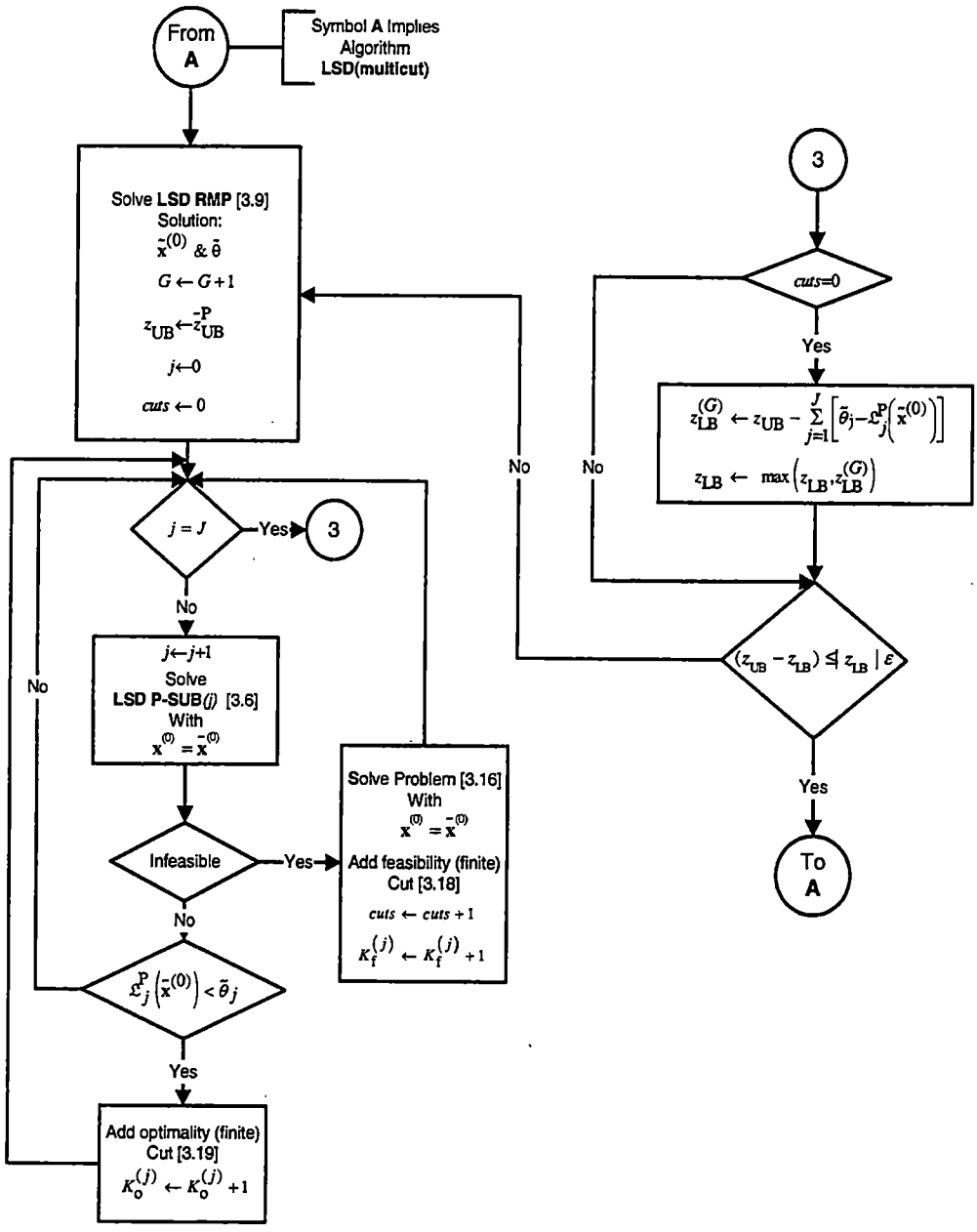


Figure 3.4: Procedure **LSD(multicut)-Optimize** Flowchart

Relaxed formulation. These five properties are stated below as Propositions 4-8 followed by Proposition 9 formally stating the finite termination property.

First, problem **LSD RMP** [3.9] is rewritten by replacing each cut set C_j, $j = 1, \ldots, J$, with an explicit listing of contained constraints and by superscribing the objective function designator with the current algorithm **LSD(multicut)** iteration index G. Let $K_o^{(j,G)}$ and $K_f^{(j,G)}$ designate the numbers of optimality and feasibility cuts respectively at the conclusion of iteration G with $K_o^{(j,0)} = K_f^{(j,0)} = 0$ for cut sets $j = 1, \ldots, J$. Finite (B) and unbounded (U) subscript identifiers are removed from the applicable dual multipliers. Then, at **LSD(multicut)** iteration $G \geq 1$, **LSD RMP**(G) is:

$$
\begin{array}{rlllll}
\tilde{z}_{UB}^{P(G)} = \max & \mathbf{c}'^{(0)}\mathbf{x}^{(0)} & + \sum_{j=1}^{J} u_j\theta_j & & & \\
\text{s.t.} & \mathbf{A}^{(0,0)}\mathbf{x}^{(0)} & & \leq & \mathbf{b}^{(0)}, & \\
& \boldsymbol{\pi}_k'^{(j)}\mathbf{A}^{(j,0)}\mathbf{x}^{(0)} & + \theta_j & \leq & \mathbf{b}'^{(j)}\boldsymbol{\pi}_k^{(j)}, & k = 1, \ldots, K_o^{(j,G-1)}, \\
& & & & & j = 1, \ldots, J, \\
& \boldsymbol{\delta}_k'^{(j)}\mathbf{A}^{(j,0)}\mathbf{x}^{(0)} & & \leq & \mathbf{b}'^{(j)}\boldsymbol{\delta}_k^{(j)}, & k = 1, \ldots, K_f^{(j,G-1)}, \\
& & & & & j = 1, \ldots, J, \\
& & \mathbf{x}^{(0)} & \geq & \mathbf{0}, & \\
& & \theta_j & \text{free}, & & j = 1, \ldots, J,
\end{array}
\qquad [3.20]
$$

The dual to problem [3.20] will also be required in the proof of the first proposition below. Let, $\boldsymbol{\pi}^{(0)}$, $\boldsymbol{\eta}^{(j)}$, and $\upsilon^{(j)}$ be the duals to the first, second, and third sets of constraints of problem [3.20] respectively. The corresponding dual problem is then:

$$
\begin{aligned}
\tilde{z}_{\text{UB}}^{\text{D}(G)} = \min \quad & \mathbf{b}'^{(0)}\boldsymbol{\pi}^{(0)} + \sum_{j=1}^{J}\sum_{k=1}^{K_{\text{o}}^{(j,G-1)}} \mathbf{b}'^{(j)}\boldsymbol{\pi}_k^{(j)}\eta_k^{(j)} \\
& + \sum_{j=1}^{J}\sum_{k=1}^{K_{\text{f}}^{(j,G-1)}} \mathbf{b}'^{(j)}\boldsymbol{\delta}_k^{(j)}\upsilon_k^{(j)} \\
\text{s.t.} \quad & \mathbf{A}'^{(0,0)}\boldsymbol{\pi}^{(0)} + \sum_{j=1}^{J}\sum_{k=1}^{K_{\text{o}}^{(j,G-1)}} \mathbf{A}'^{(j,0)}\boldsymbol{\pi}_k^{(j)}\eta_k^{(j)} \\
& + \sum_{j=1}^{J}\sum_{k=1}^{K_{\text{f}}^{(j,G-1)}} \mathbf{A}'^{(j,0)}\boldsymbol{\delta}_k^{(j)}\upsilon_k^{(j)} \geq \mathbf{c}^{(0)}, \\
& \sum_{k=1}^{K_{\text{o}}^{(j,G-1)}} \eta_k^{(j)} = u_j, \quad j = 1,\ldots,J, \\
& \boldsymbol{\pi}^{(0)} \geq \mathbf{0}, \\
& \eta_k^{(j)} \geq 0, \quad k = 1,\ldots,K_{\text{o}}^{(j,G-1)},\ j = 1,\ldots,J, \\
& \upsilon_k^{(j)} \geq 0, \quad k = 1,\ldots,K_{\text{f}}^{(j,G-1)},\ j = 1,\ldots,J.
\end{aligned}
\qquad [3.21]
$$

Define a feasible iteration of algorithm **LSD(multicut)** as one with an optimal solution to the current relaxed master problem, say $\mathbf{x}^{(0,G)}$ and $\boldsymbol{\theta}^{(G)}$. Denote the **LSD Relaxed** formulation, problems **LSD RMP**(G) [3.20] and **LSD P-SUB**(j) [3.6], at iteration G of algorithm **LSD(multicut)** as **LSD Relaxed**(G). Then, Propositions 4-9 formally state properties of the **LSD(multicut)** algorithm and the **LSD Relaxed** formulation that were implicitly assumed in the development of the algorithm.

Proposition 4 *At any feasible iteration G of algorithm* **LSD(multicut)** *with* $\mathcal{Z}_{\text{UB}}(\mathbf{x}^{(0,G)}) < \infty$, *at least one optimality cut constraint is tight in each* C_j, $j \in \mathsf{J}$, *that has* $K_{\text{o}}^{(j,G-1)} \geq 1$.

Proof Note that $\tilde{z}_{\text{UB}}^{\text{P}(G)} < \infty$ when $\mathcal{Z}_{\text{UB}}(\mathbf{x}^{(0,G)}) < \infty$ since either $u_j = 0$ or $u_j = 1$ and $K_{\text{o}}^{(j,G-1)} \geq 1 \; \forall \; j = 1, \ldots, J$. Therefore, for each $j \in \text{J}$ with $K_{\text{o}}^{(j,G-1)} \geq 1$, the dual problem [3.21] constraints

$$\begin{aligned} \sum_{k=1}^{K_{\text{o}}^{(j,G-1)}} \eta_k^{(j)} &= 1, \\ \eta_k^{(j)} &\geq 0, \quad k = 1, \ldots, K_{\text{o}}^{(j,G-1)}, \end{aligned}$$

require at least one positive $\eta_k^{(j)}$ for $1 \leq k \leq K_{\text{o}}^{(j,G-1)}$. By linear programming complementary slackness conditions, the corresponding primal problem [3.20] optimality cut constraint must be tight:

$$\pi_k^{\prime(j)} \mathbf{A}^{(j,0)} \mathbf{x}^{(0,G)} + \theta_j^{(G)} = \mathbf{b}^{\prime(j)} \pi_k^{(j)}. \blacksquare$$

Proposition 5 *At any feasible iteration G of algorithm* **LSD(multicut)**, $\theta_j^{(G)} \geq \mathfrak{L}_j^{\text{P}}(\mathbf{x}^{(0,G)})$ $\forall \; j = 1, \ldots, J$.

Proof This is clearly true if $\mathcal{Z}_{\text{UB}}(\mathbf{x}^{(0,G)})$ is unbounded since $\mathfrak{L}_j^{\text{P}}(\mathbf{x}^{(0,G)})$ is set to negative infinity for $j = 1, \ldots, J$ in Step 2 of procedure **LSD(multicut)-Initialize** – recall that $\mathcal{Z}_{\text{UB}}(\mathbf{x}^{(0)})$ is never unbounded in procedure **LSD(multicut)-Optimize**. It is also true when $\mathcal{Z}_{\text{UB}}(\mathbf{x}^{(0,G)}) < \infty$ and $u_j = 0$ for some $j \in \text{J}$ since $\theta_j^{(G)} = \infty$. Then for $\mathcal{Z}_{\text{UB}}(\mathbf{x}^{(0,G)}) < \infty$ and $u_j = 1$ for some $j \in \text{J}$ (which implies that $K_{\text{o}}^{(j,G-1)} \geq 1$), assume by contradiction that $\theta_j^{(G)} < \mathfrak{L}_j^{\text{P}}(\mathbf{x}^{(0,G)})$. The assumption is obviously not valid if $\text{X}^{(j)}\left(\mathbf{x}^{(0,G)}\right) = \emptyset$ since $\mathfrak{L}_j^{\text{P}}(\mathbf{x}^{(0,G)}) = -\infty$ by definition. Therefore, consider the case for a feasible (i.e.,

$\mathrm{X}^{(j)}\left(\mathbf{x}^{(0,G)}\right)$ is not empty) problem **LSD P-SUB(j)** [3.6] with optimal solution $\mathbf{x}^{(j,G)}$ and optimal dual multipliers $\boldsymbol{\pi}^{(j,G)}$. The assumption and strong duality imply that

$$\theta_j^{(G)} < \mathfrak{L}_j^{\mathrm{P}}(\mathbf{x}^{(0,G)}) = \mathfrak{L}_j^{\mathrm{D}}(\mathbf{x}^{(0,G)}) = \left(\mathbf{b}^{(j)} - \mathbf{A}^{(j,0)}\mathbf{x}^{(0,G)}\right)' \boldsymbol{\pi}^{(j,G)}.$$

The corresponding optimality cuts of problem **LSD RMP(G)** [3.20],

$$\boldsymbol{\pi}_k^{\prime(j)}\mathbf{A}^{(j,0)}\mathbf{x}^{(0,G)} - \theta_j^{(G)} \leq \mathbf{b}^{\prime(j)}\boldsymbol{\pi}_k^{(j)},\ k = 1, \ldots, K_{\mathrm{o}}^{(j,G-1)},$$

must also be satisfied. However, $\boldsymbol{\pi}_k^{\prime(j)}$, $k = 1, \ldots, K_{\mathrm{o}}^{(j,G-1)}$, are feasible but not necessarily optimal to problem **LSD D-SUB(j)** [3.7] for all $\mathbf{x}^{(0)} \in \mathrm{X}_j^{(0)}$, implying that

$$\theta_j^{(G)} < \left(\mathbf{b}^{(j)} - \mathbf{A}^{(j,0)}\mathbf{x}^{(0,G)}\right)' \boldsymbol{\pi}^{(j,G)} \leq \left(\mathbf{b}^{(j)} - \mathbf{A}^{(j,0)}\mathbf{x}^{(0,G)}\right)' \boldsymbol{\pi}_k^{(j)},\ k = 1, \ldots, K_{\mathrm{o}}^{(j,G-1)},$$

which violates Proposition 4. Therefore, the assumption that $\theta_j^{(G)} < \mathfrak{L}_j^{\mathrm{P}}(\mathbf{x}^{(0,G)})$ cannot be valid. ■

Proposition 6 *At any iteration G, algorithm* **LSD(multicut)** *maintains a valid lower bound, z_{LB}, on the value, z^*, of problem* **LBALP** [3.4]. *Moreover, z_{LB} is the best (greatest) lower bound found by the algorithm through iteration G.*

Proof The value z_{LB} is initially set to negative infinity and is changed thereafter to $z_{\mathrm{LB}}^{(G)} = \mathbf{c}^{\prime(0)}\mathbf{x}^{(0)} + \sum_{j=1}^{J} \mathfrak{L}_j^{\mathrm{P}}(\mathbf{x}^{(0)})$ only when $\mathbf{x}^{(j)} \in \mathrm{X}_j^{(0)}$, $j = 0, \ldots, J$ (i.e., $\mathbf{x}^{(j)}$, $j = 0, \ldots, J$, have been found that are feasible to problem **LBALP**) with $z_{\mathrm{LB}}^{(G)} > z_{\mathrm{LB}}$. ■

Proposition 7 *At any iteration G, algorithm* **LSD(multicut)** *maintains* **LSD Relaxed**(G) *as a relaxation of formulation* **LSD Master-Sub**, *problems* [3.5] and [3.6], *and thus the original problem* **LBALP** [3.4].

Proof Assume $\widetilde{\mathbf{x}}^{(j)}$, $j = 0, \ldots, J$, and $\widetilde{\theta}_j$, $j = 1, \ldots, J$, are feasible to formulation **LSD Master-Sub**. Then $\widetilde{\mathbf{x}}^{(j)}$, $j = 1, \ldots, J$, are clearly feasible to formulation **LSD Relaxed**(G) since the two formulations share the same phase two subproblems **LSD P-SUB**(j) [3.6]. The point $\widetilde{\mathbf{x}}^{(0)} \in \mathsf{X}_0^{(0)}$ in both formulations and $\widetilde{\mathbf{x}}^{(0)}$ must satisfy any feasibility cuts,

$$\delta_k'^{(j)} \mathbf{A}^{(j,0)} \widetilde{\mathbf{x}}^{(0)} \leq \mathrm{b}'^{(j)} \delta_k^{(j)}, \; k = 1, \ldots, K_{\mathrm{f}}^{(j,G-1)}, \; j = 1, \ldots, J,$$

in **LSD Relaxed**(G) since these cuts were formed to admit only $\mathbf{x}^{(0)}$ such that $\mathsf{X}^{(j)}(\mathbf{x}^{(0)})$, $j \in \mathsf{J}$, is not empty (i.e., problem **LSD P-SUB**(j) is feasible). Since the constraints (3.5c) in problem **LSD Master** [3.5] require that $\widetilde{\theta}_j \leq \mathfrak{L}_j^{\mathrm{P}}(\widetilde{\mathbf{x}}^{(0)})$, $j = 1, \ldots, J$, any optimality cuts to **LSD Relaxed**(G),

$$\pi_k'^{(j)} \mathbf{A}^{(j,0)} \widetilde{\mathbf{x}}^{(0)} + \widetilde{\theta}_j \leq \mathrm{b}'^{(j)} \pi_k^{(j)}, \; k = 1, \ldots, K_{\mathrm{o}}^{(j,G-1)}, \; j = 1, \ldots, J,$$

must also be satisfied due to $\pi_k'^{(j)}$, $k = 1, \ldots, K_{\mathrm{o}}^{(j,G-1)}$, feasible but not necessarily optimal to problem **LSD D-SUB**(j), implying that

$$\widetilde{\theta}_j \leq \mathfrak{L}_j^{\mathrm{P}}(\widetilde{\mathbf{x}}^{(0)}) = \mathfrak{L}_j^{\mathrm{D}}(\widetilde{\mathbf{x}}^{(0)}) \leq (\mathrm{b}^{(j)} - \mathbf{A}^{(j,0)} \widetilde{\mathbf{x}}^{(0)})' \pi_k^{(j)}, \; k = 1, \ldots, K_{\mathrm{o}}^{(j,G-1)}, \; j = 1, \ldots, J.$$

Therefore, any $\mathbf{x}^{(j)}$, $j = 0, \ldots, J$, and θ_j, $j = 1, \ldots, J$, feasible to **LSD Master-Sub** is also feasible to **LSD Relaxed**(G). Conversely, Proposition 5 indicates that $\mathbf{x}^{(j)}$, $j = 0, \ldots, J$, and θ_j, $j = 1, \ldots, J$, feasible to **LSD Relaxed**(G) may not be feasible to **LSD Master-Sub**

since the latter requires that $\theta_j \leq \Omega_j^{\mathrm{P}}(\mathbf{x}^{(0)})$, $j = 1, \ldots, J$. Since algorithm **LSD(multicut)** replaces the value of problem **LSD Relaxed**(G), $\tilde{z}_{\mathrm{UB}}^{\mathrm{P}(G)}$, with $z_{\mathrm{UB}} = \mathbf{c}'^{(0)}\mathbf{x}^{(0,G)} + \sum_{j=1}^{J} \theta_j^{(G)}$, Proposition 5 also implies that $z_{\mathrm{UB}} \geq z^*$. Therefore, **LSD Relaxed**(G) is a relaxation of **LSD Master-Sub** and thus the original problem **LBALP** [3.4] since the latter two are equivalent. ■

Proposition 8 *The original problem* **LBALP** [3.4] *is infeasible or unbounded if and only if algorithm* **LSD(multicut)** *terminates with an indication that the problem is infeasible or unbounded respectively.*

Proof The algorithm terminates with an indication that the problem is unbounded only when procedure **LSD(multicut)-Initialize** finds $\mathbf{x}^{(j)}$, $j = 0, \ldots, J$, feasible to problem **LBALP** such that $\mathbf{c}'^{(j)}\mathbf{x}^{(j)}$ is unbounded for at least one $j \in 0 \cap \mathrm{J}$. **LSD(multicut)** terminates with an indication of infeasible only when problem **LSD RMP**(G) [3.20] is found to be infeasible. The latter implies that problem **LBALP** is infeasible by Proposition 7. Conversely, the algorithm terminates at iteration G with an ϵ-optimal solution only if $z_{\mathrm{LB}} > -\infty$ and $z_{\mathrm{UB}} < \infty$ indicating that $\mathbf{x}^{(j)}$, $j = 0, \ldots, J$, feasible to problem **LBALP** have been found with $z_{\mathrm{LB}} \leq z^* \leq z_{\mathrm{UB}}$ by Propositions 6 and 7. ■

Proposition 9 *Algorithm* **LSD(multicut)** *terminates in a finite number of iterations either with an indication that the original problem* **LBALP** [3.4] *is infeasible or unbounded or with an ϵ-optimal solution to problem* **LBALP** *given that problems* **LSD RMP**(G) [3.20]

and **LSD P-SUB**(j) [3.6], $j = 1, \ldots, J$, *are not degenerate or that the simplex solver prevents cycling.*

Proof Cycling prevention and/or the absence of degeneracy insures that the solutions to **LSD RMP**(G) and **LSD P-SUB**(j), $j = 1, \ldots, J$, will be obtained in a finite number of simplex iterations at each iteration of algorithm **LSD(multicut)**. Each $\pi_k^{(j)}$ and $\delta_k^{(j)}$ coefficient vector in the optimality and feasibility cuts corresponds to a basis of the coefficient matrix (after adding columns to account for slack variables) of the appropriate problem [3.6], [3.10], [3.12], or [3.16]. Since each of these matrices has a finite number of square non-singular submatrices, there are a finite number of possible feasibility and optimality cuts. No $\pi_k^{(j)}$or $\delta_k^{(j)}$ will be generated more than once since any subsequent generation would imply the admission of a solution from outside the current feasible region including the cut based upon the initial generation. Therefore, algorithm **LSD(multicut)** will terminate in a finite number of iterations.

Assume that the algorithm does not terminate with an indication that the problem is infeasible or unbounded implying that problem **LBALP** is feasible and bounded by Proposition 8. The algorithm concludes each iteration, G, either by adding at least one cut and starting the next iteration or by terminating with an ϵ-optimal solution. Therefore, a sequence of non-increasing upper bounds are generated since the feasible region for **LSD Relaxed**(G) is a subset of the feasible region for **LSD Relaxed**($G-1$). Assume that the last possible cut is added during iteration $\widetilde{G}$. The algorithm will then gen-

erate **LSD RMP**$(\widetilde{G}+1)$ solution $(\mathbf{x}^{(0,\widetilde{G}+1)}, \boldsymbol{\theta}^{(\widetilde{G}+1)})$ with objective value $z_{\text{UB}} = \tilde{z}_{\text{UB}}^{\text{P}(\widetilde{G}+1)} = \mathbf{c}'^{(0)}\mathbf{x}^{(0,\widetilde{G}+1)} + \sum_{j=1}^{J} \theta_j^{(\widetilde{G}+1)}$ at iteration $\widetilde{G}+1$. Each **LSD P-SUB**(j) must be feasible with $\mathfrak{L}_j^{\text{P}}\left(\mathbf{x}^{(0,\widetilde{G}+1)}\right) = \theta_j^{(\widetilde{G}+1)}$, $j = 1, \ldots, J$, due to the assumption that the last possible cut was generated during the previous iteration implying that

$$z_{\text{LB}} = \mathbf{c}'^{(0)}\mathbf{x}^{(0,\widetilde{G}+1)} + \sum_{j=1}^{J} \mathfrak{L}_j^{\text{P}}\left(\mathbf{x}^{(0,\widetilde{G}+1)}\right) = z_{\text{UB}}.$$

Therefore, by Propositions 6-8, algorithm **LSD(multicut)** either determines that problem **LBALP** is infeasible or unbounded or it generates a sequence of upper bounds and a sequence of lower bounds that converge to an ϵ-optimal solution of problem **LBALP**. ■

3.2 Dantzig-Wolfe Decomposition

Applying L-Shaped decomposition to a given LP is equivalent to applying Dantzig-Wolfe decomposition (DWD) to the dual of that LP. Thus, LSD (DWD) can be interpreted as the dual method of DWD (LSD). The duality of the two decomposition procedures has been known since the inception of LSD (see Van Slyke and Wets [196, page 653]). For instance, the application of LSD to problem [3.1] is equivalent to the application of DWD to the dual problem:

$$\begin{array}{rlrcrcll} z^* = & \min & \mathbf{b}'^{(0)}\boldsymbol{\pi}^{(0)} & + & \mathbf{b}'^{(1)}\boldsymbol{\pi}^{(1)} & & & \text{(3.22a)} \\ & \text{s.t.} & \mathbf{A}'^{(0,0)}\boldsymbol{\pi}^{(0)} & + & \mathbf{A}'^{(1,0)}\boldsymbol{\pi}^{(1)} & \geq & \mathbf{c}^{(0)}, & \text{(3.22b)} \\ & & & & \mathbf{A}'^{(1,1)}\boldsymbol{\pi}^{(1)} & \geq & \mathbf{c}^{(1)}, & \text{(3.22c)} \\ & & & & \boldsymbol{\pi}^{(j)} & \geq & \mathbf{0}, \quad j = 0, 1. & \text{(3.22d)} \end{array} \qquad [3.22]$$

Constraints (3.22b) are known as linking constraints and constraints (3.22c) are termed linked constraints. Dantzig-Wolfe decomposition, like L-Shaped decomposition, is an iter-

ative two-phase algorithm that replaces the original problem with a restricted master problem (RMP) - subproblem formulation with a subproblem for each set of linked constraints. Solutions to the restricted master problem (phase one) and the subproblems (phase two) are obtained during each iteration of the algorithm. These results are then analyzed to determine if an ϵ-optimal solution to the original problem has been found. If an ϵ-optimal solution is not available, the solutions to the subproblems are used to modify the restricted master problem by adding one or more activities (columns - as opposed to cuts or constraints in LSD) to the restricted master problem and a new iteration is executed. The procedure continues until an ϵ-optimal solution to the original problem is found or the original problem is determined to be infeasible or unbounded.

Derivations for DWD for the case of one set of linked constraints similar to problem [3.22] are given in Bazaraa, Jarvis, and Sherali [6, Chapter 7], Chvatal [34, Chapter 26], and Nazareth [157, Chapter 12]. That version is termed the **DWD(single-activity)** algorithm since the single set of linked constraints induces the addition of at most one activity to the restricted master problem during any iteration. Dantzig and Wolfe [49] and Dantzig [43, Chapter 23] provide a derivation for the case of two sets of linked constraints but with at most one new activity added at each iteration. The algorithm developed in the remainder of this section is termed the **DWD(multiactivities)** algorithm since it applies to the case of an arbitrary (though finite) number of sets of linked constraints and allows the addition of a new activity for each set of linked constraints. Dantzig and Wolfe [50] also give a version

of the algorithm for multiple sets of linked constraints, but that version is significantly less detailed than the one presented below.

3.2.1 DWD for Multiple Sets of Linked Constraints

Algorithm **DWD(multiactivities)** is developed to obtain an ϵ-optimal solution to the dual of problem **LBALP** [3.4]. The dual problem **UBALP**:

$$
\begin{aligned}
z^* = \min \quad & \mathbf{b}'^{(0)}\boldsymbol{\pi}^{(0)} + \sum_{j=1}^{J} \mathbf{b}'^{(j)}\boldsymbol{\pi}^{(j)} && && && \text{(3.23a)}\\
\text{s.t.} \quad & \mathbf{A}'^{(0,0)}\boldsymbol{\pi}^{(0)} + \sum_{j=1}^{J} \mathbf{A}'^{(j,0)}\boldsymbol{\pi}^{(j)} && \geq && \mathbf{c}^{(0)}, && \text{(3.23b)}\\
& && \mathbf{A}'^{(j,j)}\boldsymbol{\pi}^{(j)} && \geq && \mathbf{c}^{(j)}, \quad j = 1, \ldots, J, && \text{(3.23c)}\\
& && \boldsymbol{\pi}^{(j)} && \geq && \mathbf{0}, \quad j = 0, \ldots, J, && \text{(3.23d)}
\end{aligned}
$$

[3.23]

has multiple sets of linked constraints, constraints (3.23c). The matrices of coefficients in constraints (3.23b-c) form the upper block-angular structure depicted in Figure 3.5.

Define the feasibility sets

$$\Pi^{(j)} = \left\{ \boldsymbol{\pi}^{(j)} \,\middle|\, \mathbf{A}'^{(j,j)}\boldsymbol{\pi}^{(j)} \geq \mathbf{c}^{(j)}, \boldsymbol{\pi}^{(j)} \geq \mathbf{0} \right\}, j = 1, \ldots, J,$$

and note that problem **UBALP** [3.23] is infeasible if $\Pi^{(j)} = \emptyset$ for any $j \in \mathsf{J} = \{1, \ldots, J\}$. Moreover, for each $j \in \mathsf{J}$ such that $\Pi^{(j)} \neq \emptyset$, polyhedral theory (e.g., see Theorem 2.1 of Bazaraa, Jarvis, and Sherali [6, page 69] and/or Theorem 1.3-2 of Nazareth [157, page 10]) assures that:

1. $\Pi^{(j)}$ has a nonempty, finite set of extreme points, say $\boldsymbol{\pi}_k^{(j)}$, $k = 1, \ldots, K_{\mathrm{p}}^{(j)}$,

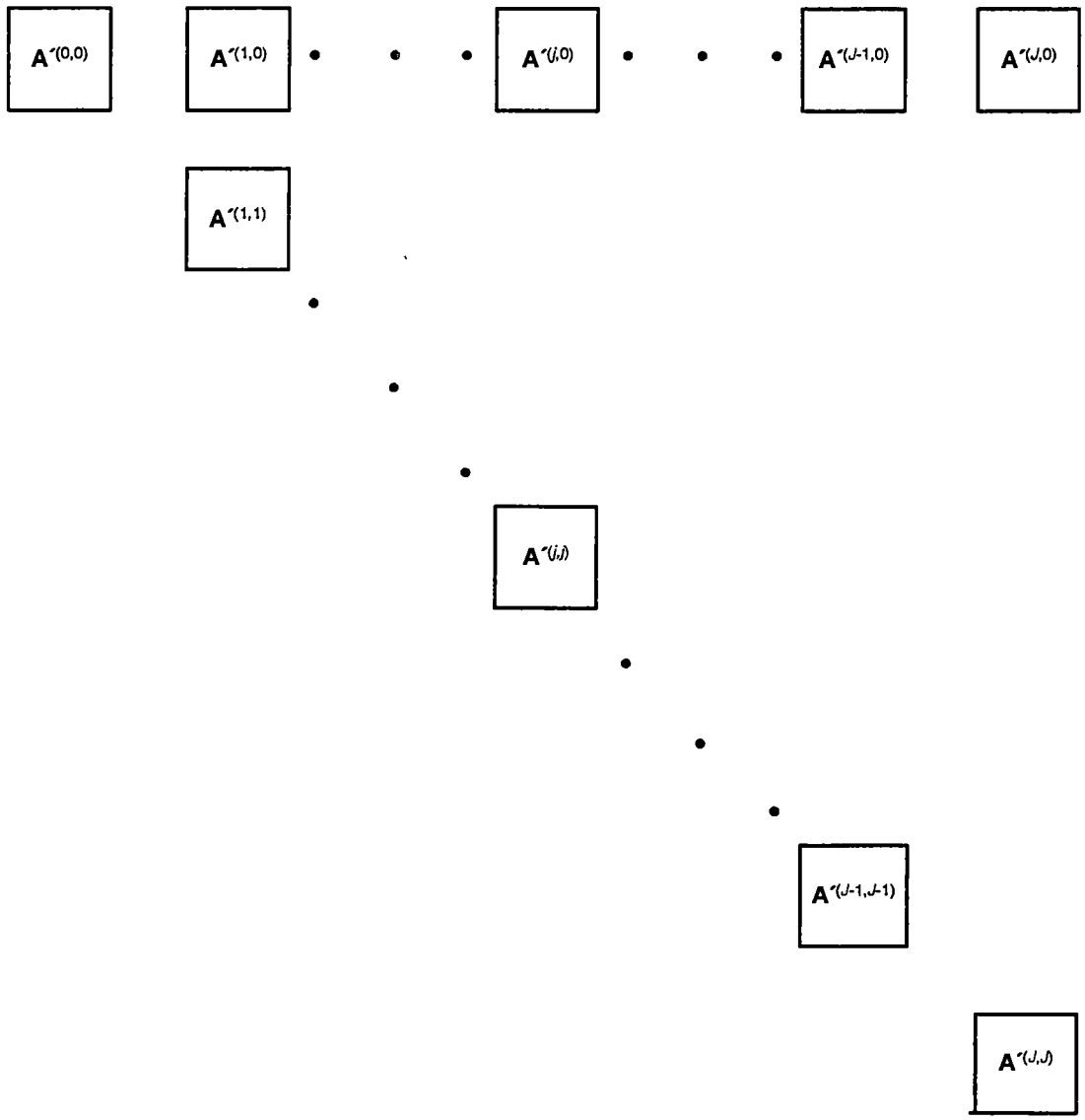


Figure 3.5: Upper Block-Angular Structure

2. $\Pi^{(j)}$ has a finite set (which is empty if and only if $\Pi^{(j)}$ is bounded) of extreme directions, say $\boldsymbol{\delta}_k^{(j)}$, $k = 1, \ldots, K_{\mathrm{d}}^{(j)}$, and

3. $\boldsymbol{\pi}^{(j)} \in \Pi^{(j)}$ if and only if

$$\begin{aligned}
\boldsymbol{\pi}^{(j)} &= \sum_{k=1}^{K_{\mathrm{p}}^{(j)}} \boldsymbol{\pi}_k^{(j)} \eta_k^{(j)} + \sum_{k=1}^{K_{\mathrm{d}}^{(j)}} \boldsymbol{\delta}_k^{(j)} \upsilon_k^{(j)}, \\
\sum_{k=1}^{K_{\mathrm{p}}^{(j)}} \eta_k^{(j)} &= 1, \\
\eta_k^{(j)} &\geq 0, k = 1, \ldots, K_{\mathrm{p}}^{(j)}, \\
\upsilon_k^{(j)} &\geq 0, k = 1, \ldots, K_{\mathrm{d}}^{(j)}.
\end{aligned}$$

Therefore, assuming that $\Pi^{(j)} \neq \emptyset$ for $j = 1, \ldots, J$, problem **UBALP** can be reformulated as the equivalent master problem, **DWD Master**:

$$\begin{aligned}
z^* = \min \quad & \mathbf{b}'^{(0)} \boldsymbol{\pi}^{(0)} + \sum_{j=1}^{J} \sum_{k=1}^{K_{\mathrm{p}}^{(j)}} \mathbf{b}'^{(j)} \boldsymbol{\pi}_k^{(j)} \eta_k^{(j)} \\
& + \sum_{j=1}^{J} \sum_{k=1}^{K_{\mathrm{d}}^{(j)}} \mathbf{b}'^{(j)} \boldsymbol{\delta}_k^{(j)} \upsilon_k^{(j)} \\
\text{s.t.} \quad & \mathbf{A}'^{(0,0)} \boldsymbol{\pi}^{(0)} + \sum_{j=1}^{J} \sum_{k=1}^{K_{\mathrm{p}}^{(j)}} \mathbf{A}'^{(j,0)} \boldsymbol{\pi}_k^{(j)} \eta_k^{(j)} \\
& + \sum_{j=1}^{J} \sum_{k=1}^{K_{\mathrm{d}}^{(j)}} \mathbf{A}'^{(j,j)} \boldsymbol{\delta}_k^{(j)} \upsilon_k^{(j)} \geq \mathbf{c}^{(0)}, \\
& \sum_{k=1}^{K_{\mathrm{p}}^{(j)}} \eta_k^{(j)} = 1, \quad j = 1, \ldots, J, \\
& \boldsymbol{\pi}^{(0)} \geq \mathbf{0}, \\
& \eta_k^{(j)} \geq 0, \quad k = 1, \ldots, K_{\mathrm{p}}^{(j)}, \; j = 1, \ldots, J, \\
& \upsilon_k^{(j)} \geq 0, \quad k = 1, \ldots, K_{\mathrm{d}}^{(j)}, \; j = 1, \ldots, J,
\end{aligned} \tag{3.24}$$

where the first set of inequality constraints are known as *base* constraints and the equality constraints are known as *convexity* constraints.

Solving problem **DWD Master** directly is generally not practicable since the extreme points and extreme directions are probably not known beforehand and since the numbers of extreme points, $K_{\mathrm{p}}^{(j)}$, and extreme directions, $K_{\mathrm{d}}^{(j)}$, for any $j \in \mathsf{J}$ are usually very large. However, assume that there are known subsets of extreme points, $\left\{\widetilde{\boldsymbol{\pi}}_1^{(j)}, \ldots, \widetilde{\boldsymbol{\pi}}_{\widetilde{K}_{\mathrm{p}}^{(j)}}^{(j)}\right\}$, and extreme directions, $\left\{\widetilde{\boldsymbol{\delta}}_1^{(j)}, \ldots, \widetilde{\boldsymbol{\delta}}_{\widetilde{K}_{\mathrm{d}}^{(j)}}^{(j)}\right\}$, with $\widetilde{K}_{\mathrm{p}}^{(j)} \geq 1$ and $\widetilde{K}_{\mathrm{d}}^{(j)} \geq 0$, for $\Pi^{(j)}$, $j = 1, \ldots, J$. Then the restricted master problem, **DWD RMP**:

$$
\begin{aligned}
z_{\mathrm{UB}}^{\mathrm{D}} = \min \quad & \mathbf{b}'^{(0)}\boldsymbol{\pi}^{(0)} + \sum_{j=1}^{J}\sum_{k=1}^{\widetilde{K}_{\mathrm{p}}^{(j)}} \mathbf{b}'^{(j)}\widetilde{\boldsymbol{\pi}}_k^{(j)}\eta_k^{(j)} \\
& + \sum_{j=1}^{J}\sum_{k=1}^{\widetilde{K}_{\mathrm{d}}^{(j)}} \mathbf{b}'^{(j)}\widetilde{\boldsymbol{\delta}}_k^{(j)}\upsilon_k^{(j)} \\
\text{s.t.} \quad & \mathbf{A}'^{(0,0)}\boldsymbol{\pi}^{(0)} + \sum_{j=1}^{J}\sum_{k=1}^{\widetilde{K}_{\mathrm{p}}^{(j)}} \mathbf{A}'^{(j,0)}\widetilde{\boldsymbol{\pi}}_k^{(j)}\eta_k^{(j)} \\
& + \sum_{j=1}^{J}\sum_{k=1}^{\widetilde{K}_{\mathrm{d}}^{(j)}} \mathbf{A}'^{(j,j)}\widetilde{\boldsymbol{\delta}}_k^{(j)}\upsilon_k^{(j)} \geq \mathbf{c}^{(0)}, \\
& \sum_{k=1}^{\widetilde{K}_{\mathrm{p}}^{(j)}} \eta_k^{(j)} = 1, \qquad j = 1, \ldots, J, \\
& \boldsymbol{\pi}^{(0)} \geq \mathbf{0}, \\
& \eta_k^{(j)} \geq 0, \qquad k = 1, \ldots, \widetilde{K}_{\mathrm{p}}^{(j)}, \; j = 1, \ldots, J, \\
& \upsilon_k^{(j)} \geq 0, \qquad k = 1, \ldots, \widetilde{K}_{\mathrm{d}}^{(j)}, \; j = 1, \ldots, J.
\end{aligned}
\tag{3.25}
$$

clearly provides an upper bound on **DWD Master** (i.e., $z_{\mathrm{UB}}^{\mathrm{D}} \geq z^*$) since any optimal solution for **DWD RMP** is feasible but not necessarily optimum to **DWD Master**. Optimality of **DWD RMP** requires that, for $j = 1, \ldots, J$, the reduced costs for all optimum extreme

point convexity parameters, say $\widetilde{\eta}_k^{(j)}$, $k = 1, \ldots, \widetilde{K}_{\mathrm{p}}^{(j)}$, and all optimum extreme direction parameters, say $\widetilde{\delta}_k^{(j)}$, $k = 1, \ldots, \widetilde{K}_{\mathrm{d}}^{(j)}$, be nonnegative. Let $\mathbf{x}^{(0)}$ and θ_j, $j = 1, \ldots, J$, be the optimum dual multipliers to **DWD RMP** and thus an optimal solution to the dual problem:

$$
\begin{array}{rlrcll}
z_{\mathrm{UB}}^{\mathrm{P}} = & \max & \mathbf{c}'^{(0)}\mathbf{x}^{(0)} + \sum_{j=1}^{J} \theta_j & & & \\
& \text{s.t.} & \mathbf{A}^{(0,0)}\mathbf{x}^{(0)} & \leq & \mathbf{b}^{(0)}, & \\
& & \widetilde{\boldsymbol{\pi}}_k'^{(j)}\mathbf{A}^{(j,0)}\mathbf{x}^{(0)} + \theta_j & \leq & \mathbf{b}'^{(j)}\widetilde{\boldsymbol{\pi}}_k^{(j)}, & k = 1, \ldots, \widetilde{K}_{\mathrm{p}}^{(j)}, \\
& & & & & j = 1, \ldots, J, \\
& & \widetilde{\boldsymbol{\delta}}_k'^{(j)}\mathbf{A}^{(j,0)}\mathbf{x}^{(0)} & \leq & \mathbf{b}'^{(j)}\widetilde{\boldsymbol{\delta}}_k^{(j)}, & k = 1, \ldots, \widetilde{K}_{\mathrm{d}}^{(j)}, \\
& & & & & j = 1, \ldots, J, \\
& & \mathbf{x}^{(0)} & \geq & \mathbf{0}, & \\
& & \theta_j & \text{free}, & & j = 1, \ldots, J.
\end{array}
\qquad [3.26]
$$

Nonnegativity of reduced costs at optimality of **DWD RMP** [3.25] implies that for $j = 1, \ldots, J$:

$$
\begin{aligned}
\mathbf{b}'^{(j)}\widetilde{\boldsymbol{\pi}}_k^{(j)} - \widetilde{\boldsymbol{\pi}}_k'^{(j)}\mathbf{A}^{(j,0)}\mathbf{x}^{(0)} - \theta_j &= \left(\mathbf{b}^{(j)} - \mathbf{A}^{(j,0)}\mathbf{x}^{(0)}\right)'\widetilde{\boldsymbol{\pi}}_k^{(j)} - \theta_j \geq 0, k = 1, \ldots, \widetilde{K}_{\mathrm{p}}^{(j)}, \\
\mathbf{b}'^{(j)}\widetilde{\boldsymbol{\delta}}_k^{(j)} - \widetilde{\boldsymbol{\delta}}_k'^{(j)}\mathbf{A}^{(j,0)}\mathbf{x}^{(0)} &= \left(\mathbf{b}^{(j)} - \mathbf{A}^{(j,0)}\mathbf{x}^{(0)}\right)'\widetilde{\boldsymbol{\delta}}_k^{(j)} \geq 0, k = 1, \ldots, \widetilde{K}_{\mathrm{d}}^{(j)}.
\end{aligned}
\qquad [3.27]
$$

Moreover, the optimal solution to **DWD RMP** is optimum to **DWD Master** only if the reduced costs associated with all extreme points and extreme directions of $\Pi^{(j)}$, $j = 1, \ldots, J$, are nonnegative. The latter condition can be verified by solving the phase two subproblems, **DWD D-SUB**(j), $j = 1, \ldots, J$:

$$
\begin{array}{rlrcl}
\mathfrak{D}_j^{\mathrm{D}}(\mathbf{x}^{(0)}) = & \min & \left(\mathbf{b}^{(j)} - \mathbf{A}^{(j,0)}\mathbf{x}^{(0)}\right)'\boldsymbol{\pi}^{(j)} & & \\
& \text{s.t.} & \mathbf{A}'^{(j,j)}\boldsymbol{\pi}^{(j)} & \geq & \mathbf{c}^{(j)}, \\
& & \boldsymbol{\pi}^{(j)} & \geq & \mathbf{0},
\end{array}
\qquad [3.28]
$$

which correspond to the dual phase two subproblems, **DWD P-SUB**(j), $j = 1, \ldots, J$:

$$\begin{aligned} \mathfrak{D}_j^{\mathrm{P}}(\mathbf{x}^{(0)}) = \quad \max \quad & \mathbf{c}'(j)\mathbf{x}^{(j)} \\ \text{s.t.} \quad & \mathbf{A}^{(j,j)}\mathbf{x}^{(j)} \leq \mathbf{b}^{(j)} - \mathbf{A}^{(j,0)}\mathbf{x}^{(0)}, \\ & \mathbf{x}^{(j)} \geq \mathbf{0}. \end{aligned} \qquad [3.29]$$

Since $\Pi^{(j)}$, $j = 1, \ldots, J$, are assumed to be nonempty, **DWD D-SUB**(j) either has a finite optimal solution at an extreme point of $\Pi^{(j)}$ or is unbounded along an extreme direction of $\Pi^{(j)}$.

A bounded **DWD D-SUB**(j) for some $j \in \mathrm{J}$ with $\mathfrak{D}_j^{\mathrm{D}}(\mathbf{x}^{(0)}) - \theta_j < 0$ implies an extreme point solution, say $\pi^{(j)} = \widetilde{\pi}^{(j)}_{\widetilde{K}_{\mathrm{p}}^{(j)}+1}$, such that

$$\left(\mathbf{b}^{(j)} - \mathbf{A}^{(j,0)}\mathbf{x}^{(0)}\right)' \widetilde{\pi}^{(j)}_{\widetilde{K}_{\mathrm{p}}^{(j)}+1} - \theta_j < 0.$$

Note that $\widetilde{\pi}^{(j)}_{\widetilde{K}_{\mathrm{p}}^{(j)}+1} \notin \left\{\widetilde{\pi}^{(j)}_1, \ldots, \widetilde{\pi}^{(j)}_{\widetilde{K}_{\mathrm{p}}^{(j)}}\right\}$ since the extreme points in the subset are included in **DWD RMP** and each is associated with a nonnegative reduced cost as shown in the first relation of [3.27]. Therefore, $\widetilde{\pi}^{(j)}_{\widetilde{K}_{\mathrm{p}}^{(j)}+1}$ is included in the subset of extreme points for $\Pi^{(j)}$ and a new activity, say $\eta^{(j)}_{\widetilde{K}_{\mathrm{p}}^{(j)}+1}$ (referred to as an extreme point activity), is eligible to enter the basis of problem **DWD RMP** [3.25] for each $j \in \mathrm{J}$ such that problem **DWD D-SUB**(j) [3.28] is bounded with $\mathfrak{D}_j^{\mathrm{D}}(\mathbf{x}^{(0)}) - \theta_j < 0$. If **DWD D-SUB**($j$) is bounded with $\mathfrak{D}_j^{\mathrm{D}}(\mathbf{x}^{(0)}) - \theta = 0$ for $j = 1, \ldots, J$, then $z_{\mathrm{UB}} = z^*$ and the current optimal solution to **DWD RMP** is also optimum to problem **DWD Master** [3.24] and thus the original problem **UBALP** [3.23].

An unbounded **DWD D-SUB**(j) for some $j \in \mathrm{J}$ implies a feasible solution, say $\pi^{(j)} = \widetilde{\delta}^{(j)}_{\widetilde{K}^{(j)}_{\mathrm{d}}+1}$, such that

$$\left(\mathrm{b}^{(j)} - \mathbf{A}^{(j,0)}\mathbf{x}^{(0)}\right)' \lambda \widetilde{\delta}^{(j)}_{\widetilde{K}^{(j)}_{\mathrm{d}}+1} < 0 \; \forall \, \lambda \geq \widetilde{\lambda}$$

for some $\widetilde{\lambda} > 0$. Note that $\widetilde{\delta}^{(j)}_{\widetilde{K}^{(j)}_{\mathrm{d}}+1} \notin \left\{\widetilde{\delta}^{(j)}_{1}, \ldots, \widetilde{\delta}^{(j)}_{\widetilde{K}^{(j)}_{\mathrm{d}}}\right\}$ since the extreme directions in the subset are included in **DWD RMP** and each is associated with a nonnegative reduced cost as shown in the second relation of [3.27]. Therefore, $\widetilde{\delta}^{(j)}_{\widetilde{K}^{(j)}_{\mathrm{d}}+1}$ is included in the subset of extreme directions for $\Pi^{(j)}$ and a new activity, say $v^{(j)}_{\widetilde{K}^{(j)}_{\mathrm{d}}+1}$ (referred to as an extreme direction activity), is eligible to enter the basis of problem **DWD RMP** [3.25] for each $j \in \mathrm{J}$ such that problem **DWD D-SUB**(j) [3.28] is unbounded.

Problems **DWD RMP** and **DWD D-SUB**(j), $j = 1, \ldots, J$, also provide a valid lower bound on any feasible original problem **UBALP** [3.23] as shown by the following proposition.

Proposition 10 *Assume problem* **UBALP** [3.23] *is feasible and let* $\widetilde{\mathrm{x}}^{(0)}$ *and* $\widetilde{\theta}_j$, $j = 1, \ldots, J$, *be optimal solutions to the dual problem* [3.26] *of the current* **DWD RMP** [3.25]. *Then,* $z^* \geq z_{\mathrm{LB}}$ *where*:

$$z_{\mathrm{LB}} = z^{\mathrm{D}}_{\mathrm{UB}} + \sum_{j=1}^{J} \left[\mathfrak{D}^{\mathrm{D}}_j(\widetilde{\mathrm{x}}^{(0)}) - \widetilde{\theta}_j\right].$$

Proof Let $\widetilde{\pi}^{(j)}$, $j = 0, \ldots, J$, be any feasible solution to problem **UBALP**. Then, $\widetilde{\pi}^{(j)}$, $j \in \mathrm{J}$, feasible but not necessarily optimum to **DWD D-SUB**(j) [3.28] implies that:

$$\left(\mathbf{b}^{(j)} - \mathbf{A}^{(j,0)}\widetilde{\mathbf{x}}^{(0)}\right)' \widetilde{\pi}^{(j)} \geq \mathfrak{D}_j^{\mathrm{D}}(\widetilde{\mathbf{x}}^{(0)}), \qquad j = 1, \ldots, J$$

$$\Longrightarrow \qquad \mathbf{b}'^{(j)}\widetilde{\pi}^{(j)} \geq \widetilde{\mathbf{x}}'^{(0)}\mathbf{A}'^{(j,0)}\widetilde{\pi}^{(j)} + \mathfrak{D}_j^{\mathrm{D}}(\widetilde{\mathbf{x}}^{(0)}), \qquad j = 1, \ldots, J$$

$$\Longrightarrow \sum_{j=1}^{J} \mathbf{b}'^{(j)}\widetilde{\pi}^{(j)} \geq \widetilde{\mathbf{x}}'^{(0)} \sum_{j=1}^{J} \mathbf{A}'^{(j,0)}\widetilde{\pi}^{(j)} + \sum_{j=1}^{J} \mathfrak{D}_j^{\mathrm{D}}(\widetilde{\mathbf{x}}^{(0)}).$$

Constraints (3.23b) of problem **UBALP** imply that:

$$\sum_{j=1}^{J} \mathbf{A}'^{(j,0)}\widetilde{\pi}^{(j)} \geq \mathbf{c}^{(0)} - \mathbf{A}'^{(0,0)}\widetilde{\pi}^{(0)}$$

$$\Longrightarrow \sum_{j=1}^{J} \mathbf{b}'^{(j)}\widetilde{\pi}^{(j)} \geq \mathbf{c}'^{(0)}\widetilde{\mathbf{x}}^{(0)} - \widetilde{\mathbf{x}}'^{(0)}\mathbf{A}'^{(0,0)}\widetilde{\pi}^{(0)} + \sum_{j=1}^{J} \mathfrak{D}_j^{\mathrm{D}}(\widetilde{\mathbf{x}}^{(0)})$$

$$\Longrightarrow \left(\mathbf{A}^{(0,0)}\widetilde{\mathbf{x}}^{(0)}\right)' \widetilde{\pi}^{(0)} + \sum_{j=1}^{J} \mathbf{b}'^{(j)}\widetilde{\pi}^{(j)} \geq \mathbf{c}'^{(0)}\widetilde{\mathbf{x}}^{(0)} + \sum_{j=1}^{J} \mathfrak{D}_j^{\mathrm{D}}(\widetilde{\mathbf{x}}^{(0)}).$$

The objective function and first set of constraints, $\mathbf{A}^{(0,0)}\mathbf{x}^{(0)} \leq \mathbf{b}^{(0)}$, to the dual problem [3.26] to the current **DWD RMP** then imply that:

$$\mathbf{b}'^{(0)}\widetilde{\pi}^{(0)} + \sum_{j=1}^{J} \mathbf{b}'^{(j)}\widetilde{\pi}^{(j)} \geq z_{\mathrm{UB}}^{\mathrm{P}} + \sum_{j=1}^{J} \left[\mathfrak{D}_j^{\mathrm{D}}(\widetilde{\mathbf{x}}^{(0)}) - \widetilde{\theta}_j\right]. \qquad [3.30]$$

Therefore, since inequality [3.30] is valid for all $\widetilde{\pi}^{(j)}$, $j = 0, \ldots, J$, feasible to problem **UBALP** and since the left-hand-side of inequality [3.30] is the objective function of **UBALP**, inequality [3.30] implies that:

$$z^* \geq z_{\mathrm{UB}}^{\mathrm{D}} + \sum_{j=1}^{J} \left[\mathfrak{D}_j^{\mathrm{D}}(\widetilde{\mathbf{x}}^{(0)}) - \widetilde{\theta}_j\right], \qquad [3.31]$$

where $z_{\mathrm{UB}}^{\mathrm{D}} = z_{\mathrm{UB}}^{\mathrm{P}}$ by strong duality. ■

3.2.2 Modifying DWD RMP With Additional Activities

The basics of modifying the **DWD RMP** by adding activities at the conclusion of each iteration of the DWD algorithm are discussed above. However, the above discussion implicitly assumed that the current **DWD RMP** has a feasible and finite optimal solution. Details of the **DWD RMP** modification are now presented including the cases of an infeasible or an unbounded restricted master problem.

Let $K_{\mathrm{p}}^{(j,G)}$ and $K_{\mathrm{d}}^{(j,G)}$ be the numbers of known extreme points, $\left\{\boldsymbol{\pi}_1^{(j)},\ldots,\boldsymbol{\pi}_{K_{\mathrm{p}}^{(j,G)}}^{(j)}\right\}$, and extreme directions, $\left\{\boldsymbol{\delta}_1^{(j)},\ldots,\boldsymbol{\delta}_{K_{\mathrm{d}}^{(j,G)}}^{(j)}\right\}$, respectively for $\Pi^{(j)}$, $j=1,\ldots,J$, at the conclusion of iteration G. Denote the restricted master problem at the beginning of iteration G as **DWD RMP**(G) and reformulate **DWD RMP**(G) as:

$$
\begin{aligned}
z_{\mathrm{UB}}^{\mathrm{D}(G)} = \min \quad & \mathbf{b}'^{(0)}\boldsymbol{\pi}^{(0)} + \sum_{j=1}^{J}\sum_{k=1}^{K_{\mathrm{p}}^{(j,G-1)}} \mathbf{b}'^{(j)}\boldsymbol{\pi}_k^{(j)}\eta_k^{(j)} \\
& + \sum_{j=1}^{J}\sum_{k=1}^{K_{\mathrm{d}}^{(j,G-1)}} \mathbf{b}'^{(j)}\boldsymbol{\delta}_k^{(j)}\upsilon_k^{(j)} \\
\text{s.t.} \quad & \mathbf{A}'^{(0,0)}\boldsymbol{\pi}^{(0)} + \sum_{j=1}^{J}\sum_{k=1}^{K_{\mathrm{p}}^{(j,G-1)}} \mathbf{A}'^{(j,0)}\boldsymbol{\pi}_k^{(j)}\eta_k^{(j)} \\
& + \sum_{j=1}^{J}\sum_{k=1}^{K_{\mathrm{d}}^{(j,G-1)}} \mathbf{A}'^{(j,0)}\boldsymbol{\delta}_k^{(j)}\upsilon_k^{(j)} \geq \mathbf{c}^{(0)}, \\
& \sum_{k=1}^{K_{\mathrm{p}}^{(j,G-1)}} \eta_k^{(j)} = u_j, \quad j=1,\ldots,J, \\
& \boldsymbol{\pi}^{(0)} \geq \mathbf{0}, \\
& \eta_k^{(j)} \geq 0, \quad k=1,\ldots,K_{\mathrm{p}}^{(j,G-1)},\ j=1,\ldots,J, \\
& \upsilon_k^{(j)} \geq 0, \quad k=1,\ldots,K_{\mathrm{d}}^{(j,G-1)},\ j=1,\ldots,J.
\end{aligned}
\qquad [3.32]
$$

with corresponding dual problem:

$$
\begin{array}{rrcll}
z_{\mathrm{UB}}^{\mathrm{P}(G)} = \max & \mathbf{c}'^{(0)}\mathbf{x}^{(0)} + \sum_{j=1}^{J} u_j\theta_j & & & \\
\text{s.t.} & \mathbf{A}^{(0,0)}\mathbf{x}^{(0)} & \leq & \mathbf{b}^{(0)}, & \\
& \boldsymbol{\pi}_k'^{(j)}\mathbf{A}^{(j,0)}\mathbf{x}^{(0)} + \theta_j & \leq & \mathbf{b}'^{(j)}\boldsymbol{\pi}_k^{(j)}, & k = 1,\ldots,K_{\mathrm{p}}^{(j,G-1)}, \\
& & & & j = 1,\ldots,J, \\
& \boldsymbol{\delta}_k'^{(j)}\mathbf{A}^{(j,0)}\mathbf{x}^{(0)} & \leq & \mathbf{b}'^{(j)}\boldsymbol{\delta}_k^{(j)}, & k = 1,\ldots,K_{\mathrm{d}}^{(j,G-1)}, \\
& & & & j = 1,\ldots,J, \\
& \mathbf{x}^{(0)} & \geq & \mathbf{0}, & \\
& \theta_j & \text{free}, & & j = 1,\ldots,J.
\end{array}
\qquad [3.33]
$$

The numbers of extreme points and extreme directions are all initially 0, $K_{\mathrm{p}}^{(j,0)} = K_{\mathrm{d}}^{(j,0)} = 0$, $j = 1,\ldots,J$, and either $K_{\mathrm{p}}^{(j,G-1)}$ or $K_{\mathrm{d}}^{(j,G-1)}$ may be increased by 1 at the conclusion of iteration G. The right-hand-sides of the convexity constraints, $\sum_{k=1}^{K_{\mathrm{p}}^{(j,G-1)}} \eta_k^{(j)} = u_j$, $j = 1,\ldots,J$, in **DWD RMP**(G) are also all initially 0 and u_j is changed to 1 when the first extreme point is added to the subset of extreme points $\Pi^{(j)}$, $j \in \mathrm{J}$.

Let a new extreme point activity, $\eta_{K_{\mathrm{p}}^{(j,G-1)}+1}^{(j)}$, $j \in \mathrm{J}$, created at the conclusion of iteration G be denoted by the vector

$$
\begin{bmatrix}
\mathbf{b}'^{(j)}\boldsymbol{\pi}_{K_{\mathrm{p}}^{(j,G-1)}+1}^{(j)} \\
\mathbf{A}'^{(j,0)}\boldsymbol{\pi}_{K_{\mathrm{p}}^{(j,G-1)}+1}^{(j)} \\
1
\end{bmatrix}
$$

of its objective function coefficient (first element) and technology matrix coefficients (remaining elements). Similarly, let the vector

$$\begin{bmatrix} \mathrm{b}'^{(j)}\boldsymbol{\delta}^{(j)}_{K_{\mathrm{d}}^{(j,G-1)}+1} \\ \mathrm{A}'^{(j,0)}\boldsymbol{\delta}^{(j)}_{K_{\mathrm{d}}^{(j,G-1)}+1} \\ 0 \end{bmatrix}$$

denote a new extreme direction activity, $v^{(j)}_{K_{\mathrm{d}}^{(j,G-1)}+1}$, $j \in \mathrm{J}$, created at the conclusion of iteration G.

Each of the two primary types of activities, extreme point and extreme direction, used to modify the restricted master problem has two subtypes depending upon the feasibility of **DWD RMP**(G). The four types of activities are termed extreme point (infeasible), extreme direction (infeasible), extreme point (feasible), and extreme direction (feasible). The applicability and derivation of each activity is discussed below:

1. **DWD RMP**(G) is infeasible: formulate the following simplex phase one type problem **DWD M-RMP**(G) based upon the current restricted master problem:

$$
\begin{array}{lllllll}
z_{\mathrm{UB}}^{\mathrm{M}(G)} = & \min & \mathbf{v} & + & \sum_{j=1}^{J} w_j & & \\
& \text{s.t.} & \mathbf{v} & + & & & \\
& & \mathbf{A}'^{(0,0)}\boldsymbol{\pi}^{(0)} & + & \sum_{j=1}^{J}\sum_{k=1}^{K_{\mathrm{p}}^{(j,G-1)}} \mathbf{A}'^{(j,0)}\boldsymbol{\pi}_k^{(j)}\eta_k^{(j)} & & \\
& & & + & \sum_{j=1}^{J}\sum_{k=1}^{K_{\mathrm{d}}^{(j,G-1)}} \mathbf{A}'^{(j,0)}\boldsymbol{\delta}_k^{(j)}\upsilon_k^{(j)} & \geq \mathbf{c}^{(0)}, & \\
& & & w_j + & \sum_{k=1}^{K_{\mathrm{p}}^{(j,G-1)}} \eta_k^{(j)} & = 1, \; j = 1,\ldots,J, & \\
& & \mathbf{v} & \geq & \mathbf{0}, & & \\
& & w_j & \geq & 0, & j = 1,\ldots,J, & \\
& & \boldsymbol{\pi}^{(0)} & \geq & \mathbf{0}, & & \\
& & \eta_k^{(j)} & \geq & 0, \; k = 1,\ldots,K_{\mathrm{p}}^{(j,G-1)}, & j = 1,\ldots,J, & \\
& & \upsilon_k^{(j)} & \geq & 0, \; k = 1,\ldots,K_{\mathrm{d}}^{(j,G-1)}, & j = 1,\ldots,J, &
\end{array}
\qquad [3.34]
$$

which is always feasible and bounded. Formulate the corresponding subproblems,

DWD DM-SUB(j), $j = 1,\ldots,J$:

$$
\begin{array}{lllll}
\mathfrak{D}_j^{\mathrm{DM}}(\mathbf{x}^{(0)}) = & \min & -\left(\mathbf{A}^{(j,0)}\mathbf{x}^{(0)}\right)'\boldsymbol{\pi}^{(j)} & & \\
& \text{s.t.} & \mathbf{A}'^{(j,j)}\boldsymbol{\pi}^{(j)} & \geq \mathbf{c}^{(j)}, & \\
& & \boldsymbol{\pi}^{(j)} & \geq \mathbf{0}, &
\end{array}
\qquad [3.35]
$$

where $\mathbf{x}^{(0)}$ is the vector of dual multipliers to the base constraints in problem **DWD M-RMP**(G) [3.34]. Solve problem **DWD M-RMP**(G) and note that $z_{\mathrm{UB}}^{\mathrm{M}(G)} > 0$ since **DWD RMP**(G) is infeasible. Let the optimal solution be:

$$
\begin{array}{ll}
\mathbf{v}^{(G)}, & \\
w_j^{(G)}, & j = 1,\ldots,J, \\
\boldsymbol{\pi}^{(0,G)}, & \\
\eta_k^{(j,G)}, & k = 1,\ldots,K_{\mathrm{p}}^{(j,G-1)}, j = 1,\ldots,J, \\
\upsilon_k^{(j,G)}, & k = 1,\ldots,K_{\mathrm{d}}^{(j,G-1)}, j = 1,\ldots,J,
\end{array}
$$

with optimal dual multipliers $\mathbf{x}^{(0,G)}$ and $\theta_j^{(G)}$, $j = 1, \ldots, J$, to the base and convexity constraints respectively. Then, for some $j \in \mathrm{J}$, solve problem **DWD DM-SUB**(j) [3.35] with $\mathbf{x}^{(0)} = \mathbf{x}^{(0,G)}$ and the type of activity to add depends upon the solution:

(a) **DWD DM-SUB**(j) is infeasible $\Longrightarrow$ stop – problem **UBALP** [3.23] is infeasible since $\Pi^{(j)} = \emptyset$.

(b) **DWD DM-SUB**(j) is feasible and bounded $\Longrightarrow$ let optimal solution be $\boldsymbol{\pi}_k^{(j)}$ where $k = K_{\mathrm{p}}^{(j,G-1)} + 1$ and add new activity based upon:

i. $\mathfrak{D}_j^{\mathrm{DM}}(\mathbf{x}^{(0,G)}) - \theta_j^{(G)} = -\left(\mathbf{A}^{(j,0)}\mathbf{x}^{(0,G)}\right)' \boldsymbol{\pi}_k^{(j)} - \theta_j^{(G)} \geq 0 \Longrightarrow$ no new activity is added since all extreme points and extreme directions have nonnegative reduced costs.

ii. $\mathfrak{D}_j^{\mathrm{DM}}(\mathbf{x}^{(0,G)}) - \theta_j^{(G)} = -\left(\mathbf{A}^{(j,0)}\mathbf{x}^{(0,G)}\right)' \boldsymbol{\pi}_k^{(j)} - \theta_j^{(G)} < 0 \Longrightarrow$ extreme point (infeasible) activity: create new extreme point activity, $\eta_k^{(j)}$, with coefficients

$$\begin{bmatrix} \mathbf{b}'^{(j)}\boldsymbol{\pi}_k^{(j)} \\ \mathbf{A}'^{(j,0)}\boldsymbol{\pi}_k^{(j)} \\ 1 \end{bmatrix}, \qquad [3.36]$$

and set $K_{\mathrm{p}}^{(j,G)} = k$.

(c) **DWD DM-SUB**(j) is feasible and unbounded $\Longrightarrow$ extreme direction (infeasible) activity: let descent direction be $\boldsymbol{\delta}_k^{(j)}$ where $k = K_{\mathrm{d}}^{(j,G-1)} + 1$ and create new

extreme direction activity, $v_k^{(j)}$, with coefficients

$$\begin{bmatrix} \mathbf{b}'^{(j)}\boldsymbol{\delta}_k^{(j)} \\ \mathbf{A}'^{(j,0)}\boldsymbol{\delta}_k^{(j)} \\ 0 \end{bmatrix}, \qquad [3.37]$$

and set $K_{\mathrm{d}}^{(j,G)} = k$.

2. **DWD RMP**(G) is feasible and bounded: Let the optimal solution be:

$$\begin{array}{ll} \boldsymbol{\pi}^{(0,G)}, & \\ \eta_k^{(j,G)}, & k = 1, \ldots, K_{\mathrm{p}}^{(j,G-1)}, j = 1, \ldots, J, \\ v_k^{(j,G)}, & k = 1, \ldots, K_{\mathrm{d}}^{(j,G-1)}, j = 1, \ldots, J, \end{array}$$

with optimal dual multipliers $\mathbf{x}^{(0,G)}$ and $\theta_j^{(G)}$, $j = 1, \ldots, J$, to the base and convexity constraints respectively. Then, for some $j \in \mathrm{J}$, solve problem **DWD D-SUB**(j) [3.28] with $\mathbf{x}^{(0)} = \mathbf{x}^{(0,G)}$ and the type of activity to add depends upon the solution:

(a) **DWD D-SUB**(j) is infeasible $\Longrightarrow$ stop – problem **UBALP** [3.23] is infeasible since $\Pi^{(j)} = \emptyset$.

(b) **DWD D-SUB**(j) is feasible and bounded $\Longrightarrow$ let optimal solution be $\boldsymbol{\pi}_k^{(j)}$ where $k = K_{\mathrm{p}}^{(j,G-1)} + 1$ and add new activity based upon:

i. $\mathfrak{D}_j^{\mathrm{D}}(\mathbf{x}^{(0,G)}) - \theta_j^{(G)} = \left(\mathbf{b}^{(j)} - \mathbf{A}^{(j,0)}\mathbf{x}^{(0,G)}\right)' \boldsymbol{\pi}_k^{(j)} - \theta_j^{(G)} \geq 0 \Longrightarrow$ no new activity is added since all extreme points and extreme directions have nonnegative reduced costs.

ii. $\mathfrak{D}_j^{\mathrm{D}}(\mathbf{x}^{(0,G)}) - \theta_j^{(G)} = \left(\mathbf{b}^{(j)} - \mathbf{A}^{(j,0)}\mathbf{x}^{(0,G)}\right)' \boldsymbol{\pi}_k^{(j)} - \theta_j^{(G)} < 0 \Longrightarrow$ extreme point (feasible) activity: create new extreme point activity, $\eta_k^{(j)}$, with coefficients

$$\begin{bmatrix} \mathbf{b}'^{(j)}\boldsymbol{\pi}_k^{(j)} \\ \mathbf{A}'^{(j,0)}\boldsymbol{\pi}_k^{(j)} \\ 1 \end{bmatrix}, \qquad [3.38]$$

and set $K_{\mathrm{p}}^{(j,G)} = k$.

(c) **DWD D-SUB**(j) is feasible and unbounded $\Longrightarrow$ extreme direction (feasible) activity: let descent direction be $\boldsymbol{\delta}_k^{(j)}$ where $k = K_{\mathrm{d}}^{(j,G-1)} + 1$ and create new extreme direction activity, $v_k^{(j)}$, with coefficients

$$\begin{bmatrix} \mathbf{b}'^{(j)}\boldsymbol{\delta}_k^{(j)} \\ \mathbf{A}'^{(j,0)}\boldsymbol{\delta}_k^{(j)} \\ 0 \end{bmatrix}, \qquad [3.39]$$

and set $K_{\mathrm{d}}^{(j,G)} = k$.

3. **DWD RMP**(G) is feasible and unbounded: stop - problem **UBALP** [3.23] is unbounded.

3.2.3 Algorithm DWD(multiactivities)

Algorithm **DWD(multiactivities)** is composed of two procedures – initialize and optimize. The initialization procedure adds at least one extreme point (feasible) type activity [3.38] to the restricted master problem for each feasibility set $\Pi^{(j)}$, $j = 1 \ldots, J$, if the procedure determines that the original problem **UBALP** [3.23] has a feasible and finite optimal solu-

tion. The optimization procedure finds an ϵ-optimal solution to problem **UBALP** given that the initialization procedure determined that a finite optimal solution exists. The algorithm is detailed below followed by the descriptions for the two contained procedures. Complementary flowcharts for the algorithm and procedures follow the detailed descriptions. Note that the algorithm assumes a predefined nonnegative value for the relative difference, ϵ, between the upper and lower bounds on the solution, z^*, to problem **UBALP**.

Algorithm DWD(multiactivities)

Step 0: Initialize the following parameters:

$$
\begin{array}{rcll}
G & \leftarrow & 0, & \\
z_{\mathrm{LB}} & \leftarrow & -\infty, & \\
z_{\mathrm{UB}} & \leftarrow & \infty & \\
u_j & \leftarrow & 0, & j = 1, \ldots, J, \\
K_{\mathrm{p}}^{(j,0)} & \leftarrow & 0, & j = 1, \ldots, J, \\
K_{\mathrm{d}}^{(j,0)} & \leftarrow & 0, & j = 1, \ldots, J, \\
feasible & \leftarrow & \text{true}, & \\
finite & \leftarrow & \text{true}, & \\
solved & \leftarrow & \text{false}. &
\end{array}
$$

Go to Step 1.

Step 1: Execute Procedure **DWD(multiactivities)-Initialize**. If $feasible$ = false: stop with an infeasible problem **UBALP** [3.23]; else if $finite$ = false: stop with an unbounded problem **UBALP**; else if $solved$ = true: stop with ϵ-optimal solution to problem **UBALP**; else go to Step 2.

Step 2: Execute Procedure **DWD(multiactivities)-Optimize**. Stop with ϵ-optimal solution to problem **UBALP**.

Parameter G is the algorithm iteration counter and is increased by one in the two referenced procedures at the beginning of each attempt to solve the current restricted master problem. The best (greatest) lower and best (least) upper bounds on the value, z^*, of problem **UBALP** found through the current iteration are recorded by z_{LB} and z_{UB} respectively. For $j = 1, \ldots, J$, the right-hand-side, u_j, of the corresponding **DWD RMP**(G) [3.32] convexity constraint is initially set to zero. The value, u_j, is set to one in procedure **DWD(multiactivities)-Initialize** when the first extreme point activity for $\Pi^{(j)}$, $j \in$ J, is added to the restricted master problem. Parameters $K_{\text{p}}^{(j,0)}$ and $K_{\text{d}}^{(j,0)}$ are set to zero to indicate that there are no extreme point or extreme direction activities in the initial restricted master problem **DWD RMP**(1). Logical parameters $feasible$, $finite$, and $solved$ indicate the return status of procedure **DWD(multiactivities)-Initialize** as follows:

$feasible$**:** state false indicates that no feasible solution exists for problem **UBALP** while state true indicates that there is a feasible solution;

$finite$**:** meaningful only when $feasible$ = true and state false means problem **UBALP** is unbounded while state true indicates that a bounded solution exists;

$solved$**:** meaningful only when $feasible = finite$ = true and state true indicates that an ϵ-optimal solution to problem **UBALP** has been determined while state false indicates that procedure **DWD(multiactivities)-Optimize** should be executed to obtain an ϵ-optimal solution.

Procedure DWD(multiactivities)-Initialize

Step 1: Set $G \leftarrow G + 1$, solve problem **DWD M-RMP**(G) [3.34] and let the optimum solution be:

$$\begin{array}{ll} \mathbf{v}^{(G)}, & \\ w_j^{(G)}, & j = 1, \ldots, J, \\ \boldsymbol{\pi}^{(0,G)}, & \\ \eta_k^{(j,G)}, & k = 1, \ldots, K_{\mathrm{p}}^{(j,G-1)}, j = 1, \ldots, J, \\ \upsilon_k^{(j,G)}, & k = 1, \ldots, K_{\mathrm{d}}^{(j,G-1)}, j = 1, \ldots, J, \end{array}$$

with optimal dual multipliers $\mathbf{x}^{(0,G)}$ and $\theta_j^{(G)}$, $j = 1, \ldots, J$, to the base and convexity constraints respectively. If $z_{\mathrm{UB}}^{\mathrm{M}(G)} > 0$ (at least one artificial variable is positive) go to Step 2; else, $z_{\mathrm{UB}}^{\mathrm{M}(G)} = 0$ (a feasible solution to problem **DWD RMP**(G) [3.32] has been found), go to Step 5.

Step 2: Set $j \leftarrow 0$, $acts \leftarrow J$ and go to Step 3.

Step 3: If $j = J$, go to Step 4. Set $j \leftarrow j + 1$, solve modified subproblem **DWD DM-SUB**(j) [3.35] with $\mathbf{x}^{(0)} = \mathbf{x}^{(0,G)}$. If **DWD DM-SUB**($j$) is infeasible, set $feasible \leftarrow$ false and return to algorithm; else, if **DWD DM-SUB**(j) is feasible and bounded, set $k = K_{\mathrm{p}}^{(j,G-1)} + 1$, let optimal solution be $\boldsymbol{\pi}_k^{(j)}$ and go to Substep 3a; else, **DWD DM-SUB**(j) is feasible and unbounded, set $k = K_{\mathrm{d}}^{(j,G-1)} + 1$, let descent direction be $\boldsymbol{\delta}_k^{(j)}$ and go to Substep 3b.

Substep 3a: If $\mathfrak{D}_j^{\mathrm{DM}}(\mathbf{x}^{(0,G)}) - \theta_j^{(G)} = -\left(\mathbf{A}^{(j,0)}\mathbf{x}^{(0,G)}\right)' \boldsymbol{\pi}_k^{(j)} - \theta_j^{(G)} \geq 0$, set $acts \leftarrow acts - 1$ and return to Step 3; else, $\mathfrak{D}_j^{\mathrm{DM}}(\mathbf{x}^{(0,G)}) - \theta_j^{(G)} = -\left(\mathbf{A}^{(j,0)}\mathbf{x}^{(0,G)}\right)' \boldsymbol{\pi}_k^{(j)} - \theta_j^{(G)} < 0$, an extreme point of $\Pi^{(j)}$ has been found with negative reduced cost, so create new extreme point (infeasible) type activity, $\eta_k^{(j)}$, with coefficients [3.36], set $K_{\mathrm{p}}^{(j,G)} = k$ and set $u_j \leftarrow 1$ if $K_{\mathrm{p}}^{(j,G)} = 1$ and return to Step 3.

Substep 3b: An extreme direction of $\Pi^{(j)}$ has been found with negative reduced cost, so create new extreme direction (infeasible) type activity, $v_k^{(j)}$, with coefficients [3.37], set $K_{\rm d}^{(j,G)} = k$ and return to Step 3.

Step 4: If $acts = 0$, set $feasible \leftarrow$ false and return to algorithm; else, $acts > 0$, return to Step 1.

Step 5: Set $z_{\rm UB}^{{\rm D}(G)}$ as follows:

$$z_{\rm UB}^{{\rm D}(G)} = {\rm b}'^{(0)}\pi^{(0,G)} + \sum_{j=1}^{J} \sum_{k=1}^{K_{\rm p}^{(j,G-1)}} {\rm b}'^{(j)}\pi_k^{(j)}\eta_k^{(j,G)} + \sum_{j=1}^{J} \sum_{k=1}^{K_{\rm d}^{(j,G-1)}} {\rm b}'^{(j)}\delta_k^{(j)}v_k^{(j,G)}.$$

If $z_{\rm UB}^{{\rm D}(G)} = -\infty$, set $bound \leftarrow$ false and return to algorithm; else, $z_{\rm UB}^{{\rm D}(G)} > -\infty$, set $z_{\rm UB} \leftarrow z_{\rm UB}^{{\rm D}(G)}$, $j \leftarrow 0$, $acts \leftarrow 0$ and go to Step 7.

Step 6: Set $G \leftarrow G + 1$, solve problem **DWD RMP**(G) [3.32]. If $z_{\rm UB}^{{\rm D}(G)} = -\infty$, set $bound \leftarrow$ false and return to algorithm; else, $z_{\rm UB}^{{\rm D}(G)} > -\infty$, set $z_{\rm UB} \leftarrow z_{\rm UB}^{{\rm D}(G)}$, and let the optimum solution be:

$$\begin{array}{ll} \pi^{(0,G)}, & \\ \eta_k^{(j,G)}, & k = 1, \ldots, K_{\rm p}^{(j,G-1)}, j = 1, \ldots, J, \\ v_k^{(j,G)}, & k = 1, \ldots, K_{\rm d}^{(j,G-1)}, j = 1, \ldots, J, \end{array}$$

with optimal dual multipliers $\mathrm{x}^{(0,G)}$ and $\theta_j^{(G)}$, $j = 1, \ldots, J$, to the base and convexity constraints respectively. Set $j \leftarrow 0$, $acts \leftarrow 0$ and go to Step 7.

Step 7: If $j = J$, go to Step 8. Set $j \leftarrow j + 1$, solve subproblem **DWD D-SUB**(j) [3.28] with $\mathrm{x}^{(0)} = \mathrm{x}^{(0,G)}$. If **DWD D-SUB**$(j)$ is bounded, set $k = K_{\rm p}^{(j,G-1)} + 1$, let optimal solution be $\pi_k^{(j)}$ and go to Substep 7a; else, **DWD D-SUB**(j) is unbounded, set $k = K_{\rm d}^{(j,G-1)} + 1$, let descent direction be $\delta_k^{(j)}$ and go to Substep 7b.

Substep 7a: Set $acts \leftarrow acts+1$. If $\mathfrak{D}_j^{\mathrm{D}}(\mathbf{x}^{(0,G)}) - \theta_j^{(G)} = \left(\mathbf{b}^{(j)} - \mathbf{A}^{(j,0)}\mathbf{x}^{(0,G)}\right)' \boldsymbol{\pi}_k^{(j)} - \theta_j^{(G)} = 0$, repeat Step 7; else, $\mathfrak{D}_j^{\mathrm{D}}(\mathbf{x}^{(0,G)}) - \theta_j^{(G)} = \left(\mathbf{b}^{(j)} - \mathbf{A}^{(j,0)}\mathbf{x}^{(0,G)}\right)' \boldsymbol{\pi}_k^{(j)} - \theta_j^{(G)} < 0$, an extreme point of $\Pi^{(j)}$ has been found with negative reduced cost, so create new extreme point (feasible) type activity, $\eta_k^{(j)}$, with coefficients [3.38], set $K_{\mathrm{p}}^{(j,G)} = k$ and return to Step 7.

Substep 7b: An extreme direction of $\Pi^{(j)}$ has been found with negative reduced cost, so create new extreme direction (feasible) type activity, $v_k^{(j)}$, with coefficients [3.39], set $K_{\mathrm{d}}^{(j,G)} = k$ and return to Step 7.

Step 8: If $acts < J$, return to Step 6. Set $z_{\mathrm{LB}}^{(G)} = z_{\mathrm{UB}} + \sum_{j=1}^{J} \left[\mathfrak{D}_j^{\mathrm{D}}(\mathbf{x}^{(0,G)}) - \theta_j^{(G)}\right]$ and then set $z_{\mathrm{LB}} \leftarrow \max\left(z_{\mathrm{LB}}, z_{\mathrm{LB}}^{(G)}\right)$. If $(z_{\mathrm{UB}} - z_{\mathrm{LB}}) \leq |z_{\mathrm{LB}}|\,\epsilon$, set *solved* $\leftarrow$ true. Return to algorithm.

Procedure **DWD(multiactivities)-Initialize** determines if problem **UBALP** [3.23] infeasible ($feasible$ = false in Step 3 or Step 4) or unbounded ($bound$ = false in Step 5 or 6) or, otherwise, is feasible and bounded ($feasible = bound$ = true). Two courses of action based upon the state of parameter *solved* are possible for the last case (feasible and bounded):

solved = **true:** an ϵ-optimal solution to problem **UBALP** has been determined and the algorithm should be terminated, otherwise,

solved = **false:** procedure **DWD(multiactivities)-Optimize** should be executed to determine an ϵ-optimal solution to problem **UBALP** – each feasibility set $\Pi^{(j)}$, $j = 1, \ldots, J$, will be represented by at least one extreme point (feasible) type activity [3.38].

Note that Steps 1-5 are not executed after the first execution of Step 7 and Steps 6-8 are executed until problem **UBALP** is determined to be feasible and bounded (i.e., a finite lower bound is established in Step 8). Problems **DWD D-SUB**(j), $j = 1, \ldots, J$, are known to be feasible upon the first entry to Step 7 since the feasible regions to problems **DWD D-SUB**(j) [3.28] and **DWD DM-SUB**(j) [3.35] are the same (i.e., $\Pi^{(j)}$) for $j = 1, \ldots, J$. In addition, for $j = 1, \ldots, J$, $\Pi^{(j)}$ is represented by at least one extreme point (infeasible) type activity [3.36] at the first entry to Step 7 (since artificial variable $w_j^{(G)} = 0$) implying that $u_j = 1$ and that $\mathfrak{D}_j^{\mathrm{D}}(\mathbf{x}^{(0,G)}) - \theta_j^{(G)} \leq 0$ in Substep 7a.

Procedure DWD(multiactivities)-Optimize

Step 1: Set $G \leftarrow G + 1$ and solve problem **DWD RMP**(G) [3.32]. Let the optimum solution be:

$$\begin{array}{ll} \boldsymbol{\pi}^{(0,G)}, & \\ \eta_k^{(j,G)}, & k = 1, \ldots, K_{\mathrm{p}}^{(j,G-1)}, j = 1, \ldots, J, \\ \upsilon_k^{(j,G)}, & k = 1, \ldots, K_{\mathrm{d}}^{(j,G-1)}, j = 1, \ldots, J, \end{array}$$

with optimal dual multipliers $\mathbf{x}^{(0,G)}$ and $\theta_j^{(G)}$, $j = 1, \ldots, J$, to the base and convexity constraints respectively. Set $z_{\mathrm{UB}} \leftarrow z_{\mathrm{UB}}^{\mathrm{D}(G)}$, $j \leftarrow 0$, $acts \leftarrow 0$ and go to Step 2.

Step 2: If $j = J$, go to Step 3. Set $j \leftarrow j + 1$, solve subproblem **DWD D-SUB**(j) [3.28] with $\mathbf{x}^{(0)} = \mathbf{x}^{(0,G)}$. If **DWD D-SUB**($j$) is bounded, set $k = K_{\mathrm{p}}^{(j,G-1)} + 1$, let

optimal solution be $\pi_k^{(j)}$ and go to Substep 2a; else, **DWD D-SUB**(j) is unbounded, set $k = K_{\mathrm{d}}^{(j,G-1)} + 1$, let descent direction be $\delta_k^{(j)}$ and go to Substep 2b.

Substep 2a: Set $acts \leftarrow acts+1$. If $\mathfrak{D}_j^{\mathrm{D}}(\mathbf{x}^{(0,G)}) - \theta_j^{(G)} = \left(\mathbf{b}^{(j)} - \mathbf{A}^{(j,0)}\mathbf{x}^{(0,G)}\right)' \pi_k^{(j)} - \theta_j^{(G)} = 0$, repeat Step 2; else, $\mathfrak{D}_j^{\mathrm{D}}(\mathbf{x}^{(0,G)}) - \theta_j^{(G)} = \left(\mathbf{b}^{(j)} - \mathbf{A}^{(j,0)}\mathbf{x}^{(0,G)}\right)' \pi_k^{(j)} - \theta_j^{(G)} < 0$, an extreme point of $\Pi^{(j)}$ has been found with negative reduced cost, so create new extreme point (feasible) type activity, $\eta_k^{(j)}$, with coefficients [3.38], set $K_{\mathrm{p}}^{(j,G)} = k$ and return to Step 2.

Substep 2b: An extreme direction of $\Pi^{(j)}$ has been found with negative reduced cost, so create new extreme direction (feasible) type activity, $\upsilon_k^{(j)}$, with coefficients [3.39], set $K_{\mathrm{d}}^{(j,G)} = k$ and return Step 2.

Step 3: If $acts = J$: set $z_{\mathrm{LB}}^{(G)} \leftarrow z_{\mathrm{UB}} + \sum_{j=1}^{J} \left[\mathfrak{D}_j^{\mathrm{D}}(\mathbf{x}^{(0,G)}) - \theta_j^{(G)}\right]$ and then set $z_{\mathrm{LB}} \leftarrow \max\left(z_{\mathrm{LB}}, z_{\mathrm{LB}}^{(G)}\right)$. If $(z_{\mathrm{UB}} - z_{\mathrm{LB}}) \leq |z_{\mathrm{LB}}|\,\epsilon$, return to algorithm; else return to Step 1.

Procedure **DWD(multiactivities)-Optimize** is executed only if procedure **DWD(multiactivities)-Initialize** determines that problem **UBALP** [3.23] is feasible and bounded. Therefore, problem **DWD RMP**(G) [3.32] is feasible and bounded and problems **DWD D-SUB**(j) [3.28], $j = 1, \ldots, J$, are feasible at every iteration G of the optimization procedure. This procedure must then return an ϵ-optimal solution for problem **UBALP** as shown in the next subsection.

Flowcharts of algorithm **DWD(multiactivities)** and the initialization and optimization procedures are at Figures 3.6-3.8.

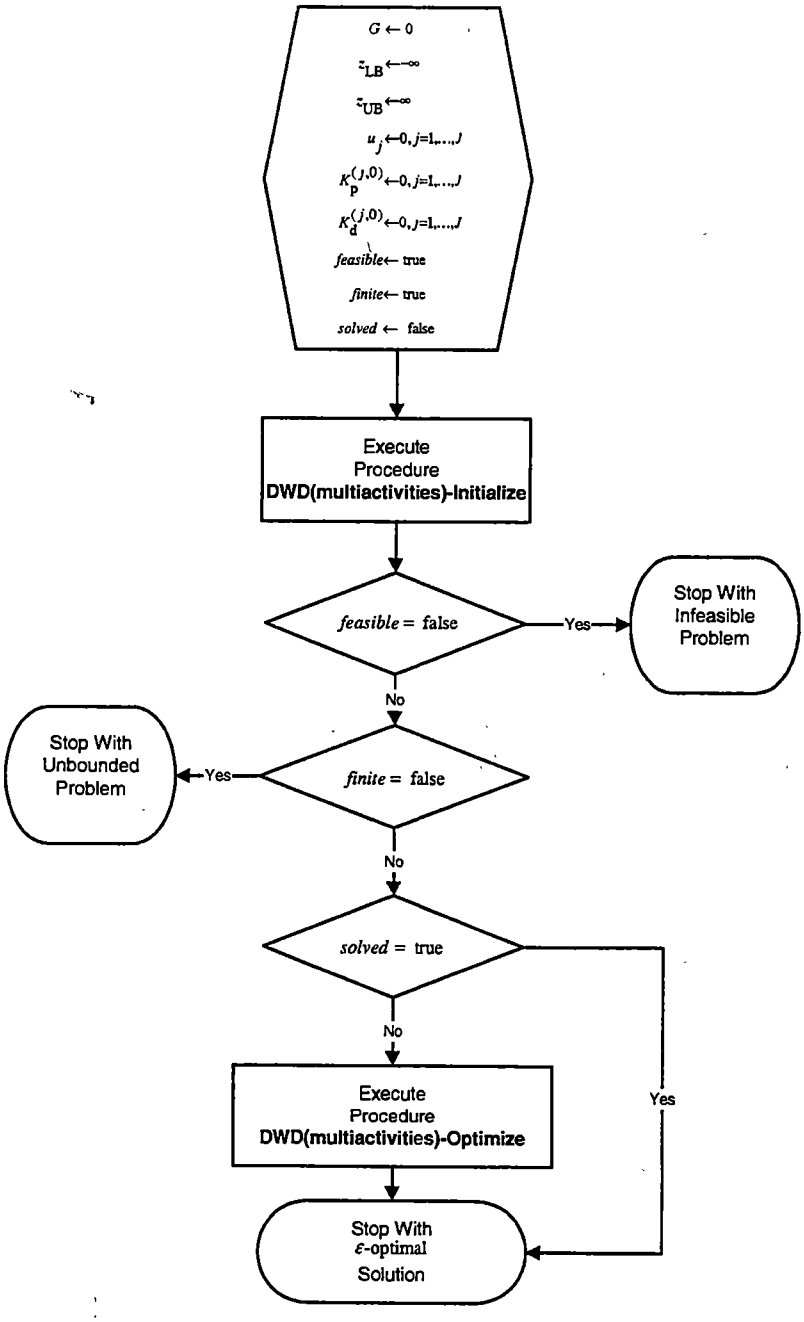


Figure 3.6: Algorithm **DWD(multiactivities)** Flowchart

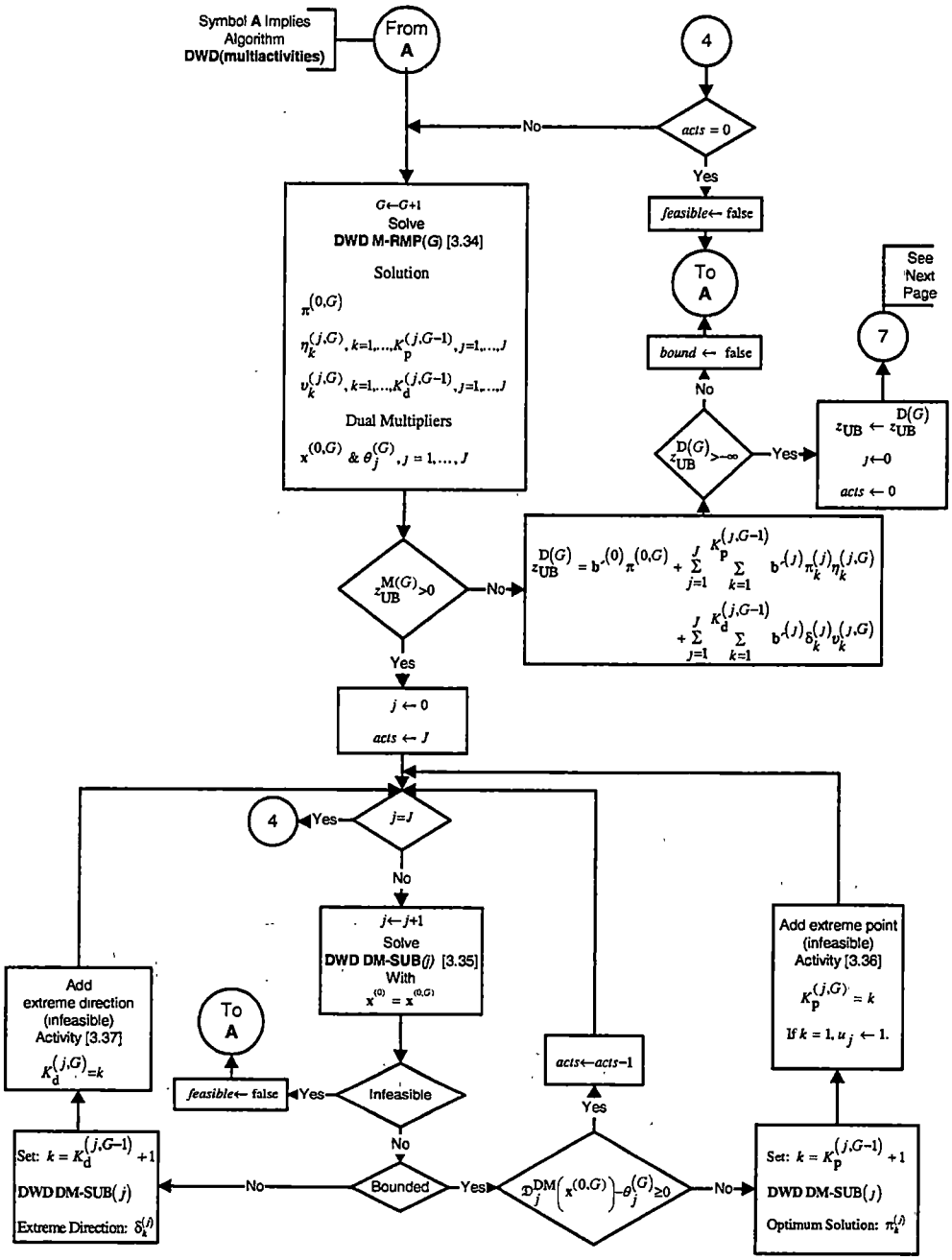


Figure 3.7: Procedure **DWD(multiactivities)-Initialize** Flowchart
(continued on next page)

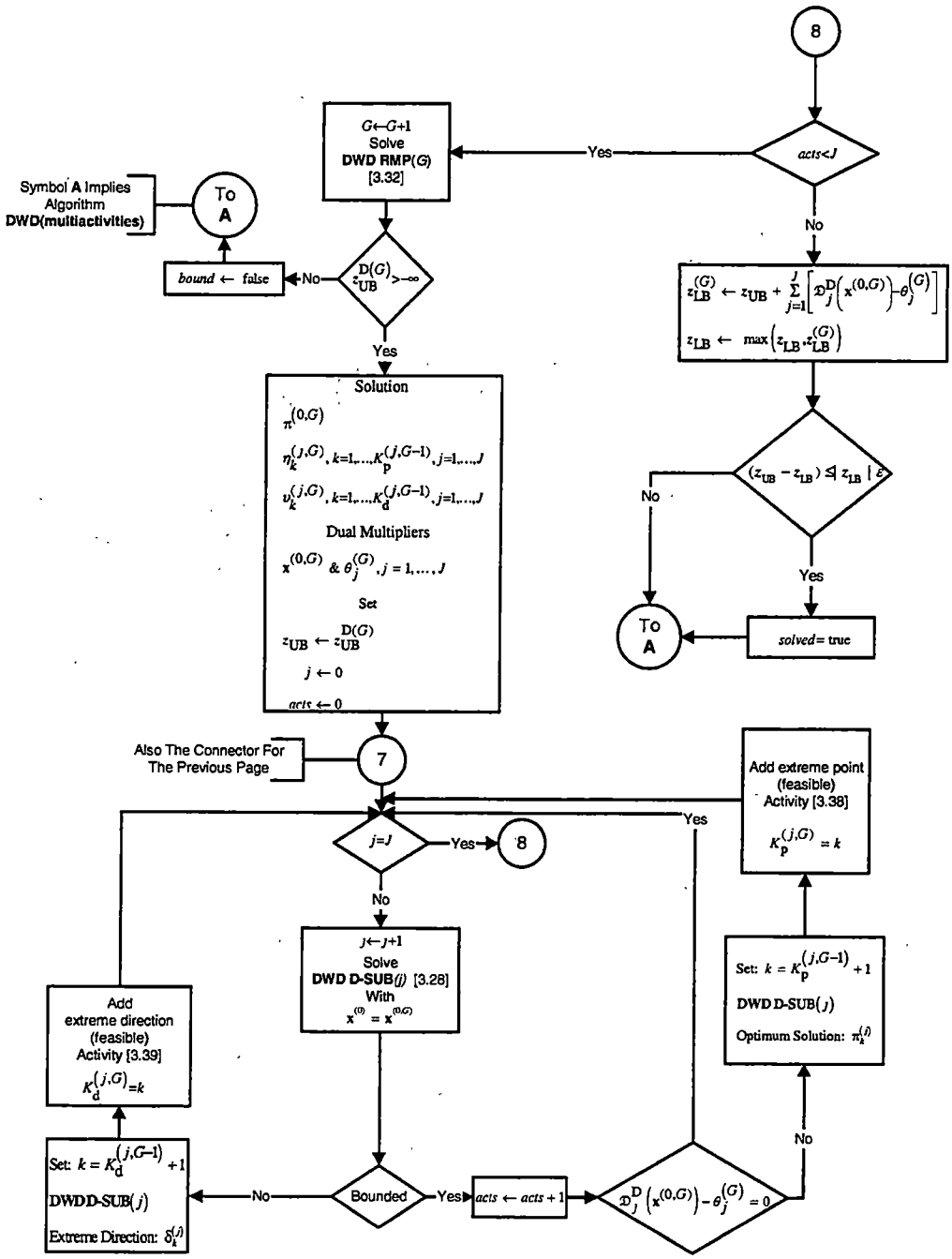


Figure 3.7: (continued) Procedure **DWD(multiactivities)-Initialize** Flowchart

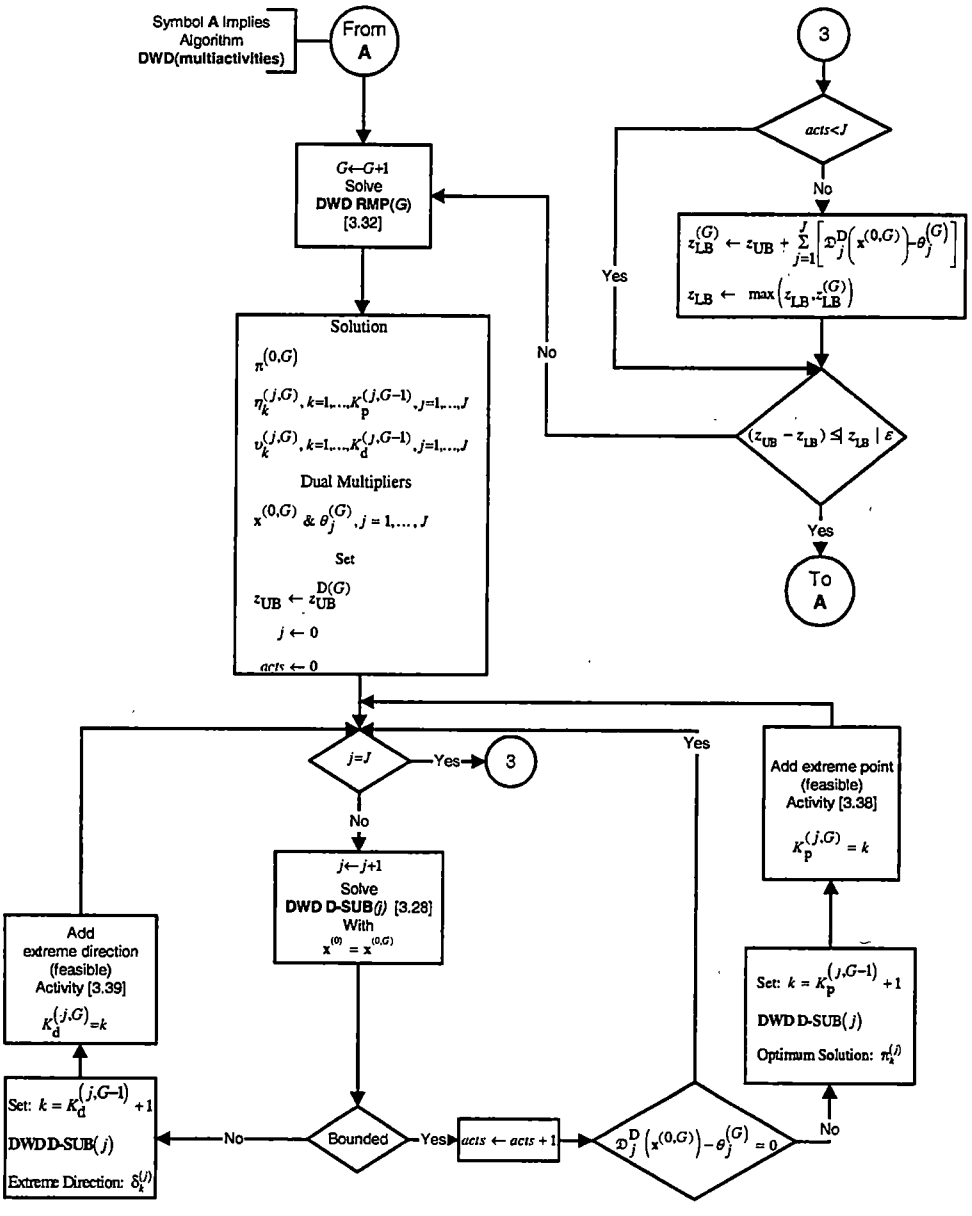


Figure 3.8: Procedure **DWD(multiactivities)-Optimize** Flowchart

3.2.4 Finite Termination of Algorithm DWD(multiactivities)

Algorithm **DWD(multiactivities)** will terminate in a finite number of iterations either with an ϵ-optimal solution to problem **UBALP** [3.23] or with an indication that problem **UBALP** is infeasible or unbounded. The finite termination property is stated formally after the following proposition concerning the feasibility and finiteness of the original problem.

Proposition 11 *The original problem* **UBALP** [3.23] *is infeasible or unbounded if and only if algorithm* **DWD(multiactivities)** *terminates with an indication that the problem is infeasible or unbounded respectively.*

Proof The algorithm terminates at iteration G with an indication that the problem is unbounded only if procedure **DWD(multiactivities)-Initialize** (Step 5 or 6) finds $\pi^{(0)}$ and extreme points $\pi_k^{(j)}$, $k = 1, \ldots, K_p^{(j,G-1)}$, and extreme directions $\delta_k^{(j)}$, $k = 1, \ldots, K_d^{(j,G-1)}$, of the feasibility sets $\Pi^{(j)}$, $j = 1, \ldots, J$, that are feasible to problem **DWD RMP**(G) [3.32] such that **DWD RMP**(G) is unbounded. Problem **DWD Master** [3.24], and thus the equivalent problem **UBALP** [3.23], is then also unbounded since any solution feasible to **DWD RMP**(G) is feasible to **DWD Master**. **DWD(multiactivities)** terminates at iteration G with an indication of infeasibility only if the initialization procedure determines infeasibility at Step 3 or Step 4. This occurs at Step 3 only if $\Pi^{(j)} = \emptyset$ for some $j \in \mathsf{J}$ indicating that **DWD Master** and **UBALP** are infeasible. Infeasibility is indicated at Step 4 only if the optimal solution to the simplex phase one type problem **DWD M-RMP**(G) [3.34] has

at least one positive artificial variable and no extreme points or extreme directions of $\Pi^{(j)}$, $j = 1, \ldots, J$, are eligible to enter the basis implying that **DWD Master**, and thus **UBALP**, is infeasible. Conversely, the algorithm terminates at iteration G with an ϵ-optimal solution only if the initialization procedure (Step 5 or 6) or the optimization procedure (Step 1) finds $\pi^{(0)}$ and extreme points $\pi_k^{(j)}$, $k = 1, \ldots, K_{\mathrm{p}}^{(j,G-1)}$, and extreme directions $\delta_k^{(j)}$, $k = 1, \ldots, K_{\mathrm{d}}^{(j,G-1)}$, of the feasibility sets $\Pi^{(j)}$, $j = 1, \ldots, J$, that are optimum to problem **DWD RMP**(G) with $-\infty < z_{\mathrm{LB}} \leq z_{\mathrm{UB}} < \infty$. This implies that **DWD Master**, and thus **UBALP**, is feasible and bounded with $z^* \leq z_{\mathrm{UB}}$ since any optimal solution to **DWD RMP**(G) is feasible but not necessarily optimum to **DWD Master**. ■

Proposition 12 *Algorithm* **DWD(multiactivities)** *terminates in a finite number of iterations either with an indication that the original problem* **UBALP** [3.23] *is infeasible or unbounded or with an ϵ-optimal solution to problem* **UBALP** *given that problems* **DWD RMP** [3.25] (*or* **DWD RMP**(G) [3.32]), **DWD M-RMP** [3.34], **DWD P-SUB**(j) [3.28], *and* **DWD PM-SUB**(j) [3.35], $j = 1, \ldots, J$, *are not degenerate or that the simplex solver prevents cycling.*

Proof Cycling prevention and/or the absence of degeneracy insures that the solutions to each applicable LP will be obtained in a finite number of simplex iterations at each iteration of algorithm **DWD(multiactivities)**. Since there are a finite number of extreme points and extreme directions for each feasibility set $\Pi^{(j)}$, $j = 1, \ldots, J$, there are a finite number of possible extreme point and extreme direction activities that can be added to the restricted

master problem. No extreme point, $\pi_k^{(j)}$, or extreme direction, $\delta_k^{(j)}$, will be generated more than once since those already generated are in the restricted master problem corresponding to activities with nonnegative reduced costs. Therefore, algorithm **DWD(multiactivities)** will terminate in a finite number of iterations.

Assume that the algorithm does not terminate with an indication that problem **UBALP** is infeasible or unbounded implying that the problem is feasible and bounded by Proposition 11. The algorithm concludes each iteration, G, either by adding at least one activity and starting the next iteration or by terminating with an ϵ-optimal solution. Therefore, a sequence of non-increasing upper bounds are generated since an optimal solution for **DWD RMP**($G-1$) is feasible but not necessarily optimum to **DWD RMP**(G). Assume that the last possible activity is added during iteration $\widetilde{G}$. Problem **DWD RMP**($\widetilde{G}+1$) is then problem **DWD Master** [3.24] so that the solution at iteration $\widetilde{G}+1$ is the solution to problem **UBALP**. Furthermore, since optimality implies nonnegative reduced costs, the solutions of problems **DWD D-SUB**(j) [3.28], $j=1,\ldots,J$, must yield

$$\mathfrak{D}_j^{\mathrm{D}}\left(\mathbf{x}^{(0,\widetilde{G}+1}\right)-\theta_j^{(\widetilde{G}+1)}=0$$

where $\mathbf{x}^{(0,\widetilde{G}+1}$and $\theta_j^{(\widetilde{G}+1)}$, $j=1,\ldots,J$, are the optimal dual multipliers to the base and convexity constraints respectively of problem **DWD RMP**($\widetilde{G}+1$) [3.32]. Proposition 10 then implies that $z_{\mathrm{LB}}=z_{\mathrm{UB}}$ at iteration $\widetilde{G}+1$. Therefore, algorithm **DWD(multiactivities)** either determines that problem **UBALP** is infeasible or unbounded or it generates a sequence

of upper bounds and a sequence of lower bounds that converge to an ϵ-optimal solution of problem **UBALP**. ■

3.3 Implementation Issues

Several implementation issues need to be addressed when considering the use of one of the decomposition techniques examined in this chapter. Seven of the more important issues are:

1. the decision to use decomposition on an LP or to solve the LP in its natural, or *grand* LP (GLP), formulation;
2. the choice between the Dantzig-Wolfe and L-Shaped decomposition algorithms;
3. the possible availability of more efficient initialization procedures for the decomposition algorithms;
4. the number of subproblems to employ in the decomposition procedure;
5. the removal of cuts (LSD) that are not tight or of extreme points/directions (DWD) that are not basic - referred to simply as removing *inactive additions*;
6. the use of a *greedy* approach, when there are multiple subproblems, in which only a subset of subproblems are solved during some iterations;
7. the accuracy of the solution, i.e., find an exact, $\epsilon = 0$, or approximate, $\epsilon > 0$, solution.

Each of the above issues is discussed in a problematic framework below. Empirical information is given on each issue in Chapter 6 with the application of decomposition to a stochastic programming formulation of a market investment model.

3.3.1 Grand LP Versus Decomposition

Decomposition may be used on any linear program although such use may require extensive array partitioning and/or artificial variables and constraints (e.g., see Chvatal [34, page 425] and Nazareth [157, Section 12.1]). Problem decomposition is most effective when the LP naturally exhibits, or is easily transformed to, one of the special, lower or upper, block-angular structures illustrated by problems **LBALP** [3.4] and **UBALP** [3.23]. Solving the problem as a GLP or by using decomposition is not always a clear-cut decision even for linear programs having a special structure. The method that provides a solution with the available resources in the shortest period of time is generally the desired option. However, the problem (or some instances of the problem) may be so large that a grand LP approach is not practicable. Decomposition procedures are particularly attractive in such cases because they generally require significantly less of the active memory of a computer than do GLP solution approaches. A major disadvantage of decomposition methods is that they usually require a significant amount of computer programming especially considering that they are most effective when tailored to a specific problem or class of problems. This disadvantage may be outweighed by the possible greater efficiency of decomposition methods especially if problem instances must be frequently solved over an extended period of time.

The option of using either or both the grand LP method and decomposition procedures for a given problem or problem class may be desirable for several reasons, including:

1. one approach may be more efficient for a particular range of problem sizes while the other approach is more effective for problems outside that range;
2. multiple solutions for the same problem obtained with different methods provide valuable code debugging and validity verification information; and
3. there may not be enough prior problem-specific information on solution times and memory requirements to make an educated judgement concerning procedural efficiencies.

3.3.2 Dantzig-Wolfe Versus L-Shaped Decomposition

The choice between DWD and LSD seems, at first glance, to be more of a straightforward decision than does the choice between decomposition and a grand LP. L-Shaped decomposition appears to be the better choice given a problem with the lower block-angular structure of problem **LBALP** [3.4]. Dantzig-Wolfe decomposition, on the other hand, seems to be the obvious choice given the upper block-angular structure of problem **UBALP** [3.23]. However, the dualistic relationships between the two problems and between the two procedures adds some uncertainty to the decision making process. It is not necessarily clear beforehand that applying LSD (DWD) to problem **LBALP** (**UBALP**) would be more efficient than applying DWD (LSD) to the dual problem **UBALP** (**LBALP**). The decision is

often to choose the decomposition procedure that has the fewest number of rows in the initial RMP (e.g., see Birge and Louveaux [16, page 385] and [17, pages 176 and 243]). For instance, apply LSD to problem **LBALP** if matrix $\mathbf{A}^{(0,0)}$ in problems [3.4] and [3.23] has more columns than rows; otherwise, apply DWD to problem **UBALP**.

L-Shaped decomposition is generally used for stochastic linear programs because the two-period problem with a discrete probability distribution has a lower block-angular structure and the desired results are the primal variables of this structure – see Birge [11, page 992] and Gassmann [83, page 408]. The preferred variables should not be a major consideration, however, if the LP solver in use gives ready access to both primal and dual variables since the primal variables in one decomposition procedure are the dual variables in the other procedure. This primal-dual relationship between the variables in the two procedures is obvious when the DWD restricted master problem and subproblems are compared with the complementary LSD relaxed master problem and subproblems. The dual problem [3.33] to problem **DWD RMP**(G) [3.32] is problem **LSD RMP**(G) [3.28] and the dual problems [3.29] to problems **DWD D-SUB**(j) [3.28] are problems **LSD P-SUB**(j) [3.6], $j = 1, \ldots, J$.

Note, however, that the two decomposition methods are theoretically indistinguishable when applied to the same problem (primal formulation for one method and dual for the other) if both the primal and dual simplex algorithms are available (see Birge and Louveaux [17, page 243]). For instance, solving the DWD restricted master problem with the primal simplex algorithm should give the same results at each simplex iteration as solv-

ing the LSD relaxed master problem with the dual simplex algorithm. This equivalence may not actually occur in practice due to particular implementations of the primal and dual simplex algorithms by the LP solver in use. Empirical evidence that one decomposition method may be more efficient than the other for a given class of problems even when both simplex algorithms are available is given in Chapter 6. The time required to solve subproblems is not a factor because both decomposition methods share the same subproblems in practice – problems **LSD D-SUB**(j) [3.7] which are the same as problems **DWD D-SUB**(j) [3.28], $j = 1, \ldots, J$.

3.3.3 Algorithm Initialization

The efficiency of either decomposition algorithm can be significantly increased if the need to execute the corresponding initialization procedure can be eliminated. Each initialization procedure has a two-fold purpose: to determine if the original problem is feasible and bounded and to create at least one optimality (finite) cut (LSD) or extreme point (feasible) activity (DWD) from each subproblem given bounded feasibility. The original problem may be known, or easily determined, to be feasible and bounded. Decomposition can be initiated in such cases by creating initial cuts/activities.

Assume that problem **LBALP** [3.4], for example, is known to be feasible and bounded. Then it might be possible to initiate algorithm **LSD(multicut)** in either of two ways. First, upper bounds can be placed on each θ_j if scalar M_j are known for **LSD P-SUB**(j) [3.6], $j = 1, \ldots, J$, such that $\mathfrak{L}_j^{\mathrm{P}}\left(\mathbf{x}^{(0)}\right) \leq M_j$ for all $\mathbf{x}^{(0)} \in \mathsf{X}_0^{(0)}$ implying that $\theta_j \leq M_j$.

Second, optimality (finite) cuts can be created for $j = 1, \ldots, J$ if some $\mathbf{x}^{(0)} \in \mathsf{X}_0^{(0)}$ can be determined such that $\mathbf{c}'^{(0)}\mathbf{x}^{(0)} < \infty$ and problems **LSD P-SUB(j)** are feasible with $\mathfrak{L}_j^{\mathrm{P}}\left(\mathbf{x}^{(0)}\right) < \infty$. Algorithm **LSD(multicut)** can then begin in the optimization procedure. Similar comments apply to problem **UBALP** [3.23] and algorithm **DWD(multiactivities)**.

3.3.4 Number of Subproblems

Problems **LBALP** [3.4] and **UBALP** [3.23] can be solved by incorporating less than J subproblems. For instance, define matrix $\widehat{\mathbf{A}}^{(1,0)}$ and block-diagonal matrix $\widehat{\mathbf{A}}^{(1,1)}$ as

$$\widehat{\mathbf{A}}^{(1,0)} = \begin{bmatrix} \mathbf{A}^{(1,0)} \\ \vdots \\ \mathbf{A}^{(j,0)} \\ \vdots \\ \mathbf{A}^{(J,0)} \end{bmatrix} \quad \text{and} \quad \widehat{\mathbf{A}}^{(1,1)} = \begin{bmatrix} \mathbf{A}^{(1,1)} & & & & \\ & \ddots & & & \\ & & \mathbf{A}^{(j,j)} & & \\ & & & \ddots & \\ & & & & \mathbf{A}^{(J,J)} \end{bmatrix},$$

and vectors $\widehat{\mathbf{b}}^{(1)}$, $\widehat{\mathbf{c}}^{(1)}$ and $\widehat{\mathbf{x}}^{(1)}$ as

$$\widehat{\mathbf{b}}^{(1)} = \begin{bmatrix} \mathbf{b}^{(1)} \\ \vdots \\ \mathbf{b}^{(j)} \\ \vdots \\ \mathbf{b}^{(J)} \end{bmatrix}, \quad \widehat{\mathbf{c}}^{(1)} = \begin{bmatrix} \mathbf{c}^{(1)} \\ \vdots \\ \mathbf{c}^{(j)} \\ \vdots \\ \mathbf{c}^{(J)} \end{bmatrix} \quad \text{and} \quad \widehat{\mathbf{x}}^{(1)} = \begin{bmatrix} \mathbf{x}^{(1)} \\ \vdots \\ \mathbf{x}^{(j)} \\ \vdots \\ \mathbf{x}^{(J)} \end{bmatrix}.$$

Then problem **LBALP** can be reformulated as the L-Shaped problem:

$$\begin{array}{rlrcl} z^* = & \max & \mathbf{c}'^{(0)}\mathbf{x}^{(0)} + & \widehat{\mathbf{c}}'^{(1)}\widehat{\mathbf{x}}^{(1)} & \\ & \text{s.t.} & \mathbf{A}^{(0,0)}\mathbf{x}^{(0)} & \leq & \mathbf{b}^{(0)}, \\ & & \widehat{\mathbf{A}}^{(1,0)}\mathbf{x}^{(0)} + & \widehat{\mathbf{A}}^{(1,1)}\widehat{\mathbf{x}}^{(1)} \leq & \widehat{\mathbf{b}}^{(1)}, \\ & & & \mathbf{x}^{(0)} \geq & \mathbf{0}, \\ & & & \widehat{\mathbf{x}}^{(1)} \geq & \mathbf{0}, \end{array}$$

with one set of linked variables, $\widehat{\mathbf{x}}^{(1)}$. Problem **UBALP** can be reformulated in a similar way to a problem with only one set of linked constraints. The reformulated problems can

then be solved with one subproblem in each case. Note that the resulting subproblem, say

$$\begin{aligned} \widehat{\mathfrak{L}}^{\mathrm{P}}(\mathbf{x}^{(0)}) = \quad \max \quad & \widehat{\mathbf{c}}'^{(1)}\widehat{\mathbf{x}}^{(1)} \\ \text{s.t.} \quad & \widehat{\mathbf{A}}^{(1,1)}\widehat{\mathbf{x}}^{(1)} \leq \widehat{\mathbf{b}}^{(1)} - \widehat{\mathbf{A}}^{(1,0)}\mathbf{x}^{(0)}, \\ & \widehat{\mathbf{x}}^{(1)} \geq \mathbf{0}, \end{aligned}$$

using LSD as an example, is separable into the original subproblems **LSD P-SUB**(j) [3.6], $j = 1, \ldots, J$. The effect of executing algorithm **LSD(single-cut)** on the reformulated problem would then be similar to executing algorithm **LSD(multicut)** on problem **LBALP** but creating one aggregated cut (either an optimality or a feasibility cut) at each iteration. In the latter case, a feasibility cut is created if any subproblem is infeasible and all other subproblems are ignored; otherwise, all subproblems are feasible and the individual optimality cuts are aggregated (summed) into one constraint with $\widehat{\theta}$ replacing $\sum_{j=1}^{J} \theta_j$. Clearly, problem **LBALP** can be formulated as a problem with $\widehat{J}$ subproblems where $1 \leq \widehat{J} \leq J$. Similar comments apply to problem **UBALP** and DWD.

Birge and Louveaux [16] and Gassmann [83] report mixed results between single-cut and multicut LSD algorithms applied to relatively small (few thousand or less constraints and variables) stochastic linear programs – some problems were solved faster with a single-cut algorithm while a multicut algorithm worked faster on other problems. Results given in Chapter 6 indicate that there may be an optimum number of subproblems for a multicut algorithm applied to an appropriate class of problems (i.e., multiple sets of linked variables or constraints).

3.3.5 Removing Inactive Additions

Murty [155] proved that optimality and feasibility cuts that are not tight can be removed from the relaxed master problem at any iteration of the LSD algorithm. A similar result holds for extreme points and extreme directions that are not in the basis of the restricted master problem in DWD. Reducing the size of the RMP is an obvious advantage of removing inactive additions. A major disadvantage, however, is that removed cuts/activities may be a part of the optimal solution and will need to be regenerated and reinserted into the RMP. There is no known method for predicting which additions are required for optimality so that any removal scheme is likely to increase the number of iterations required for convergence. Wittrock [207, page 83] proposes a procedure for removing an inactive cut in LSD based upon the number of consecutive iterations for which the cut is not tight. Removing inactive additions is not examined in detail due to reasons to be explained in Chapter 6.

3.3.6 Greedy Algorithms

All subproblems do not need to be solved at each iteration of either decomposition algorithm once the algorithm enters the applicable optimization procedure. Assume that only $\widehat{J}$, $1 \leq \widehat{J} \leq J$, subproblems are solved at some iteration $G > \widetilde{G}$ where $\widetilde{G}$ is the index for the last iteration of the applicable initialization procedure. The relaxed/restricted master problem at iteration $G+1$, say RMP($G+1$), contains the same cuts/activities present in RMP(G) plus the additional $\widehat{J}$ cuts/activities created at iteration G. Then an improved (in the ab-

sence of degeneracy) upper bound will be determined from the solution to RMP($G + 1$) since, for example, $\widehat{J}$ activities with negative reduced costs at iteration G have been added to the problem. Note, however, that a lower bound cannot be determined and used to update the best lower bound at any iteration in either algorithm unless all subproblems are solved.

This idea is particularly appealing if some subproblems are simple (e.g., can be solved without a simplex or other complex algorithm) while the others are hard (require a complex algorithm). Simple subproblems are solved at every iteration to obtain improved upper bounds in a greedy manner. Hard subproblems are solved only intermittently to update both bounds. Variations of this procedure are discussed in Chapter 6.

3.3.7 Solution Accuracy

Bazaraa, Jarvis, and Sherali [6, Section 7.1] and Nazareth [157, Section 12.2-2] indicate that decomposition algorithms often converge fairly rapidly to a neighborhood of an optimum solution but then require a substantial number of iterations to isolate the solution. Results are presented in Chapter 6 showing the time required for convergence as a function of the relative difference, ϵ, between the lower and upper bounds.

Chapter 4
Decomposition of Stochastic Linear Programs

This chapter describes the application of the decomposition techniques discussed in the previous chapter to stochastic linear programs as defined in Chapter 2. Application of Dantzig-Wolfe and L-Shaped decomposition to two-period, multi-period, and block-separable problems are covered in the first three sections below. Implementation strategies for DWD and LSD are considered in the fourth section. A third decomposition technique, referred to as *myopic* decomposition, is introduced in the fifth section. Stochastic data storage and retrieval is the topic of the sixth and final section.

4.1 Two-Period Problems

Two-period stochastic linear programs have a problem structure that is ideal for the application of DWD or LSD. Each second period node in the SLP is complementary to a phase two subproblem in either decomposition method whereas data at the first period node forms the RMP for either method. Therefore the RMP can be augmented during each algorithm iteration with a cut (LSD) or extreme point/direction activity (DWD) for each node in the second period or with aggregated cuts/activities for selected node groupings. This symmetry between DWD and LSD and two-period stochastic linear programs is a primary reason that the two-period problem is the most studied version of stochastic programs.

4.2 Multi-Period Problems – Nested Decomposition

The most common solution procedures for multi-period stochastic linear programs are based upon the structure of the corresponding decision tree. Each node in periods before the last can be visualized as the root or anchor node for a unique two-period problem where each post first-period problem appears to be *nested* within the problem at the parent node. Figure 4.1 demonstrates this concept on a four-period binary outcomes decision tree. The view from node $(1,1) = [\,]$ inside the first period problem box shows two macro-nodes represented by the second period problem boxes – one for each possible first period outcome. A similar view exists at each node in periods two and three. Therefore the overall multi-period problem can be treated conceptually as a sequence of nested two-period problems.

The nested two-period problem concept can be translated into decomposition procedures appropriately named *nested decomposition*. Dantzig and Wolfe [49, page 110] coined the term nested to describe their original outline of this technique. Dantzig [43, Section 23-4] subsequently gave a more detailed description of nested decomposition based upon DWD. Other early work using DWD was performed by Glassey [90] and Ho and Manne [102]. Nested decomposition algorithms using LSD are given in Birge [11], Birge, Donohue, Holmes, and Svintsitski [14], Birge and Louveaux [17, Section 7.1], Gassmann [82], [83], [84], and Wittrock [207].

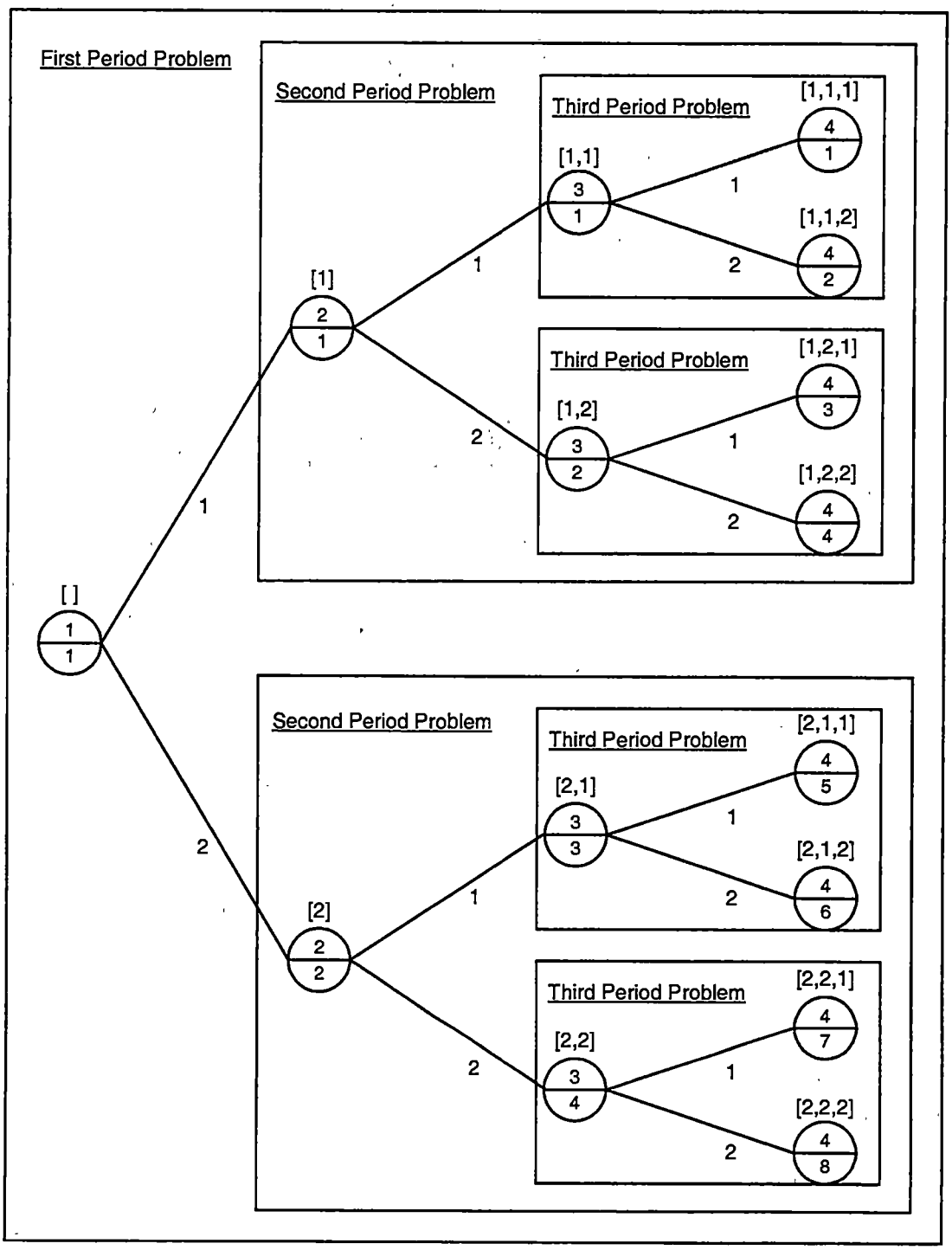


Figure 4.1: Four-Period Binary Outcomes Decision Tree - Nested Two-Period Problems

The formulations for the relaxed/restricted master problems and subproblems used in nested decomposition are given below followed by detailed descriptions of the applicable cuts and extreme points/directions. This section concludes with an examination of the procedures employed by nested decomposition algorithms.

4.2.1 RMP/SUB Formulations

Figure 4.1 suggests that nested decomposition will involve a series of RMP and SUB problems. The data for the first period node inside the first period problem box form the initial RMP for the overall problem. Data for the eight fourth period nodes are used for the pair of initial subproblems for each of the four third period nodes. Nodes in the intermediate second and third periods are then used to form problems that share both responsibilities – as a subproblem of the parent node in the previous period and as a relaxed/restricted master problem for child nodes in the next period. An intermediate t-period node acting as a SUB receives solution information from its parent node and returns cuts/activities information to the parent. Conversely, an intermediate t-period node acting as a RMP passes solution information to its child nodes and receives cuts/activities information from the children. Problems at the nodes in intermediate periods are then referred to as RMP-SUB problems. The first period node acts only as a RMP passing solution information to and receiving cuts/activities information from its child nodes. A terminal period node performs only the SUB function receiving solution information from its parent node and passing

cuts/activities information to the parent. Figure 4.2 illustrates this work flow for a generic multi-period problem.

Formulations for the RMP/SUB problems are developed for LSD using the primal multi-period problem defined in Chapter 2. Complementary formulations for DWD of the dual problem are defined at each step. These formulations are based upon representing the nodal value functions $\mathfrak{Q}^{[\bullet]_t}\left(\mathbf{x}^{[\bullet]_{t-1}}\right)$ [2.10] on page 39 for $t = 2, \ldots, T-1$ with the equivalent forms:

$$\begin{aligned}
\mathfrak{Q}^{[\bullet]_t}\left(\mathbf{x}^{[\bullet]_{t-1}}\right) = p_{l_{t-1}}^{[\bullet]_{t-1}} \max \quad & \mathbf{c}'^{[\bullet]_t}\mathbf{x}^{[\bullet]_t} + \sum_{l_t=1}^{L_t} \theta^{[\bullet]_{t+1}} && \text{(4.1a)}\\
\text{s.t.} \quad & \mathbf{W}^{[\bullet]_t}\mathbf{x}^{[\bullet]_t} && \leq \mathbf{b}^{[\bullet]_t} - \mathbf{B}^{[\bullet]_t}\mathbf{x}^{[\bullet]_{t-1}}, && \text{(4.1b)}\\
& -\mathfrak{Q}^{[\bullet]_{t+1}}\left(\mathbf{x}^{[\bullet]_t}\right) + \theta^{[\bullet]_{t+1}} && \leq 0, \; l_t = 1, \ldots, L_t, && \text{(4.1c)}\\
& \mathbf{x}^{[\bullet]_t} \geq \mathbf{0}, && && \text{(4.1d)}\\
& \theta^{[\bullet]_{t+1}} \text{ free}, \; l_t = 1, \ldots, L_t. && && \text{(4.1e)}
\end{aligned}$$

[4.1]

Constraints (4.1c) are then replaced by a set of optimality and feasibility cuts for each child node in period $t+1$ in order to define the approximate nodal value function, $\mathfrak{L}^{[\bullet]_t}\left(\mathbf{x}^{[\bullet]_{t-1}}\right)$, $t = 2, \ldots, T-1$, for problem **LSD RMP-SUB**($[\bullet]_t$):

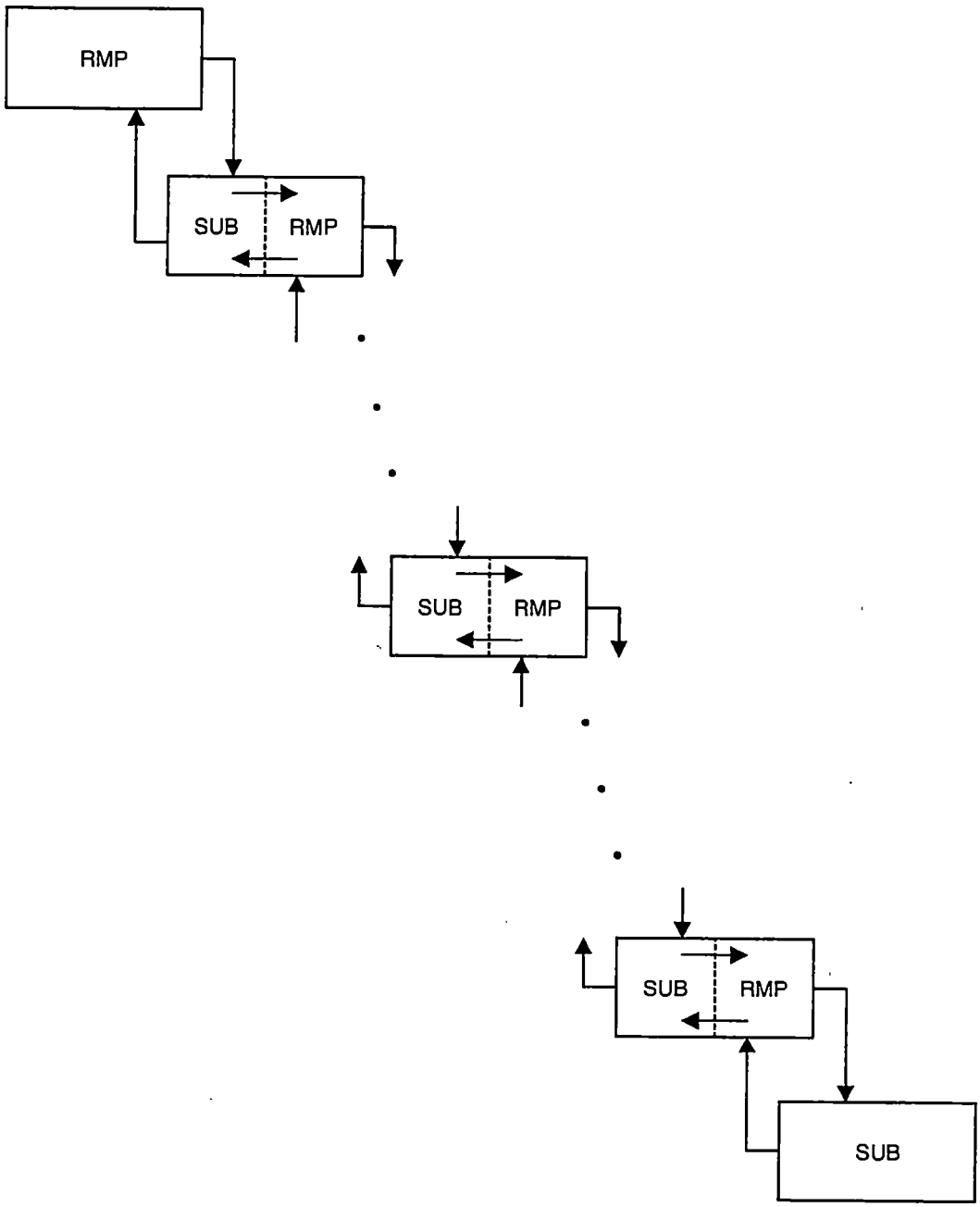


Figure 4.2: Nested Decomposition Work Flow – First Period RMP $\rightleftarrows$ Final Period SUB

$$\mathfrak{L}^{[\bullet]_t}\left(\mathbf{x}^{[\bullet]_{t-1}}\right) = p_{l_{t-1}}^{[\bullet]_{t-1}} \max \quad \mathbf{c}'^{[\bullet]_t}\mathbf{x}^{[\bullet]_t} + \sum_{l_t=1}^{L_t} \theta^{[\bullet]_{t+1}}$$

$$\begin{aligned}
\text{s.t.} \quad & \mathbf{W}^{[\bullet]_t}\mathbf{x}^{[\bullet]_t} && \leq \mathbf{b}^{[\bullet]_t} - \mathbf{B}^{[\bullet]_t}\mathbf{x}^{[\bullet]_{t-1}}, \\
& \boldsymbol{\pi}_k'^{[\bullet]_{t+1}}\mathbf{B}^{[\bullet]_{t+1}}\mathbf{x}^{[\bullet]_t} + \theta^{[\bullet]_{t+1}} && \leq b_{1k}^{[\bullet]_{t+1}}, \quad \begin{matrix} k = 1, \ldots, K_1^{[\bullet]_{t+1}}, \\ l_t = 1, \ldots, L_t, \end{matrix} \\
& \boldsymbol{\delta}_k'^{[\bullet]_{t+1}}\mathbf{B}^{[\bullet]_{t+1}}\mathbf{x}^{[\bullet]_t} && \leq b_{2k}^{[\bullet]_{t+1}}, \quad \begin{matrix} k = 1, \ldots, K_2^{[\bullet]_{t+1}}, \\ l_t = 1, \ldots, L_t, \end{matrix} \\
& \mathbf{x}^{[\bullet]_t} \geq \mathbf{0}, \\
& \theta^{[\bullet]_{t+1}} \text{ free}, \; l_t = 1, \ldots, L_t.
\end{aligned} \qquad [4.2]$$

Coefficient vectors, $\boldsymbol{\pi}_k^{[\bullet]_{t+1}}$ and $\boldsymbol{\delta}_k^{[\bullet]_{t+1}}$, and right-hand-sides, $b_{1k}^{[\bullet]_{t+1}}$ and $b_{2k}^{[\bullet]_{t+1}}$, for the optimality and feasibility cuts are defined in the following two subsections to insure that $\mathfrak{Q}^{[\bullet]_t}\left(\mathbf{x}^{[\bullet]_{t-1}}\right) \leq \mathfrak{L}^{[\bullet]_t}\left(\mathbf{x}^{[\bullet]_{t-1}}\right)$.

The overall relaxed master problem at the first period node, based upon the deterministic equivalent problem **PMPDEP** [2.11] on page 39 and referred to as **LSD RMP**([ ]), is:

$$
\begin{aligned}
z^{\mathrm{P}}_{\mathrm{UB}} = \max \quad & \mathbf{c}'^{[]}\mathbf{x}^{[]} + \sum_{l_1=1}^{L_1} \theta^{[\bullet]_2} \\
\text{s.t.} \quad & \mathbf{A}\mathbf{x}^{[]} && \leq \mathbf{b}^{[]}, \\
& \boldsymbol{\pi}_k'^{[\bullet]_2}\mathbf{B}^{[\bullet]_2}\mathbf{x}^{[]} + \theta^{[\bullet]_2} && \leq b_{1k}^{[\bullet]_2}, \quad \begin{matrix} k = 1, \ldots, K_1^{[\bullet]_2}, \\ l_1 = 1, \ldots, L_1, \end{matrix} \\
& \boldsymbol{\delta}_k'^{[\bullet]_2}\mathbf{B}^{[\bullet]_2}\mathbf{x}^{[]} && \leq b_{2k}^{[\bullet]_2}, \quad \begin{matrix} k = 1, \ldots, K_2^{[\bullet]_2}, \\ l_1 = 1, \ldots, L_1, \end{matrix} \\
& \mathbf{x}^{[]} \geq \mathbf{0}, \\
& \theta^{[\bullet]_2} \text{ free}, \; l_1 = 1, \ldots, L_1.
\end{aligned}
\qquad [4.3]
$$

Subproblems at all nodes in the terminal period, **LSD SUB**$([\bullet]_T)$, are simply the final period nodal value functions [2.9] on page 38 reproduced below for convenience:

$$
\begin{aligned}
\mathfrak{L}^{[\bullet]_T}\left(\mathbf{x}^{[\bullet]_{T-1}}\right) = p_{l_{T-1}}^{[\bullet]_{T-1}} \max \quad & \mathbf{c}'^{[\bullet]_T}\mathbf{x}^{[\bullet]_T} \\
\text{s.t.} \quad & \mathbf{W}^{[\bullet]_T}\mathbf{x}^{[\bullet]_T} \leq \mathbf{b}^{[\bullet]_T} - \mathbf{B}^{[\bullet]_T}\mathbf{x}^{[\bullet]_{T-1}}, \\
& \mathbf{x}^{[\bullet]_T} \geq \mathbf{0},
\end{aligned}
\qquad [4.4]
$$

with function identifier $\mathfrak{Q}^{[\bullet]_T}$ replaced by $\mathfrak{L}^{[\bullet]_T}$.

Restricted master problems and subproblems for DWD are the duals to the complementary problems above. The dual of problem **LSD RMP**([ ]) [4.3] is the overall restricted

master problem, **DWD RMP**([ ]):

$$
\begin{aligned}
z^{\mathrm{D}}_{\mathrm{UB}} = \min \quad & \mathbf{b}^{[]}\boldsymbol{\pi}^{[]} + \sum_{l_1=1}^{L_1}\sum_{k=1}^{K_1^{[\bullet]_2}} b_{1k}^{[\bullet]_2}\eta_k^{[\bullet]_2} + \sum_{l_1=1}^{L_1}\sum_{k=1}^{K_2^{[\bullet]_2}} b_{2k}^{[\bullet]_2}\upsilon_k^{[\bullet]_2} \\
\text{s.t.} \quad & \mathbf{A}'\boldsymbol{\pi}^{[]} + \sum_{l_1=1}^{L_1}\sum_{k=1}^{K_1^{[\bullet]_2}} \mathbf{B}'^{[\bullet]_2}\boldsymbol{\pi}_k^{[\bullet]_2}\eta_k^{[\bullet]_2} + \sum_{l_1=1}^{L_1}\sum_{k=1}^{K_2^{[\bullet]_2}} \mathbf{B}'^{[\bullet]_2}\boldsymbol{\delta}_k^{[\bullet]_2}\upsilon_k^{[\bullet]_2} \geq \mathbf{c}^{[]}, \\
& \sum_{k=1}^{K_1^{[\bullet]_2}} \eta_k^{[\bullet]_2} = 1,\ l_1 = 1,\ldots,L_1, \\
& \boldsymbol{\pi}^{[]} \geq \mathbf{0}, \\
& \eta_k^{[\bullet]_2} \geq 0, \quad \begin{array}{l} k = 1,\ldots,K_1^{[\bullet]_2}, \\ l_1 = 1,\ldots,L_1, \end{array} \\
& \upsilon_k^{[\bullet]_2} \geq 0, \quad \begin{array}{l} k = 1,\ldots,K_2^{[\bullet]_2}, \\ l_1 = 1,\ldots,L_1. \end{array}
\end{aligned}
\tag{4.5}
$$

Let the dual multipliers to the base constraints (the first set above) at any node $[\bullet]_t$ be $\mathbf{x}^{[\bullet]_t}$ and the dual multipliers to the convexity constraints (the second set) at that node be $\theta_{l_t}^{[\bullet]_{t+1}}$, $l_t = 1,\ldots,L_t$. The duals to problems **LSD RMP-SUB**($[\bullet]_t$) [4.2] at nodes in intermediate periods, $2 \leq t \leq t-1$, are then problems **DWD RMP-SUB**($[\bullet]_t$):

$$\mathfrak{D}^{[\bullet]_t}\left(\mathbf{x}^{[\bullet]_{t-1}}\right) = p_{l_{t-1}}^{[\bullet]_{t-1}} \min \quad \left(\mathbf{b}^{[\bullet]_t} - \mathbf{B}^{[\bullet]_t}\mathbf{x}^{[\bullet]_{t-1}}\right)' \boldsymbol{\pi}^{[\bullet]_t} + \sum_{l_t=1}^{L_t} \sum_{k=1}^{K_1^{[\bullet]_{t+1}}} b_{1k}^{[\bullet]_{t+1}} \eta_k^{[\bullet]_{t+1}} + \sum_{l_t=1}^{L_t} \sum_{k=1}^{K_2^{[\bullet]_{t+1}}} b_{2k}^{[\bullet]_{t+1}} \upsilon_k^{[\bullet]_{t+1}}$$

$$\text{s.t.} \quad \mathbf{W}'^{[\bullet]_t}\boldsymbol{\pi}^{[\bullet]_t} + \sum_{l_t=1}^{L_t} \sum_{k=1}^{K_1^{[\bullet]_{t+1}}} \mathbf{B}'^{[\bullet]_{t+1}}\boldsymbol{\pi}_k^{[\bullet]_{t+1}}\eta_k^{[\bullet]_{t+1}} + \sum_{l_t=1}^{L_t} \sum_{k=1}^{K_2^{[\bullet]_{t+1}}} \mathbf{B}'^{[\bullet]_{t+1}}\boldsymbol{\delta}_k^{[\bullet]_{t+1}}\upsilon_k^{[\bullet]_{t+1}} \geq \mathbf{c}^{[\bullet]_t},$$

$$\sum_{k=1}^{K_1^{[\bullet]_{t+1}}} \eta_k^{[\bullet]_{t+1}} = 1, \; l_t = 1, \ldots, L_t,$$

$$\boldsymbol{\pi}^{[\bullet]_t} \geq \mathbf{0},$$

$$\eta_k^{[\bullet]_{t+1}} \geq 0, \quad \begin{array}{l} k = 1, \ldots, K_1^{[\bullet]_{t+1}}, \\ l_t = 1, \ldots, L_t, \end{array}$$

$$\upsilon_k^{[\bullet]_{t+1}} \geq 0, \quad \begin{array}{l} k = 1, \ldots, K_2^{[\bullet]_{t+1}}, \\ l_t = 1, \ldots, L_t, \end{array}$$

[4.6]

and the duals to problems **LSD SUB**$([\bullet]_T)$ [4.4] at nodes in the terminal period are then problems **DWD SUB**$([\bullet]_T)$:

$$\begin{array}{rrcl} \mathfrak{D}^{[\bullet]_T}\left(\mathbf{x}^{[\bullet]_{T-1}}\right) = p_{l_{T-1}}^{[\bullet]_{T-1}} \min & \left(\mathbf{b}^{[\bullet]_T} - \mathbf{B}^{[\bullet]_T}\mathbf{x}^{[\bullet]_{T-1}}\right)' \boldsymbol{\pi}^{[\bullet]_T} & & \\ \text{s.t.} & \mathbf{W}'^{[\bullet]_T}\boldsymbol{\pi}^{[\bullet]_T} & \geq & \mathbf{c}^{[\bullet]_T}, \\ & \boldsymbol{\pi}^{[\bullet]_T} & \geq & \mathbf{0}. \end{array} \qquad [4.7]$$

Procedures to determine the optimality cuts (extreme point activities) and feasibility cuts (extreme direction activities) are described in the following two subsections.

4.2.2 Optimality Cuts – Extreme Point Activities

Derivations are given for the optimality cuts of L-Shaped decomposition of the primal problem with the corresponding translation to extreme point activities for Dantzig-Wolfe decomposition of the dual problem. Recall that the RMP is augmented with a new optimality cut whenever a subproblem is solved to optimality. Assume that problem **LSD RMP-SUB**$([\bullet]_t)$ [4.2] with $\mathbf{x}^{[\bullet]_{t-1}} = \tilde{\mathbf{x}}^{[\bullet]_{t-1}}$ at a node in an intermediate period, $2 \leq t \leq T-1$, is solved at some LSD iteration. Then the vector of coefficients, $\pi^{[\bullet]_t}_{K_1^{[\bullet]_t}+1}$, and the right-hand-side, $b^{[\bullet]_t}_{1\left(K_1^{[\bullet]_t}+1\right)}$, for the new optimality cut for the RMP at the parent node in period $t-1$ must be determined. Since problem **LSD RMP-SUB**$([\bullet]_t)$ has an optimal solution, the corresponding dual problem **DWD RMP-SUB**$([\bullet]_t)$ [4.6] also has an optimal solution. Let

$$\hat{\pi}^{[\bullet]_t},$$

$$\hat{\eta}^{[\bullet]_{t+1}}_{k_t}, \quad k_t = 1, \ldots, K_1^{[\bullet]_{t+1}}, \quad l_t = 1, \ldots, L_t,$$

$$\hat{v}^{[\bullet]_{t+1}}_{k_t}, \quad k_t = 1, \ldots, K_2^{[\bullet]_{t+1}}, \quad l_t = 1, \ldots, L_t,$$

be the optimal solution to the dual problem. Then, by strong duality,

$$\mathfrak{L}^{[\bullet]_t}\left(\tilde{\mathbf{x}}^{[\bullet]_{t-1}}\right) = p^{[\bullet]_{t-1}}_{l_{t-1}} \left[\begin{array}{r} \left(\mathbf{b}^{[\bullet]_t} - \mathbf{B}^{[\bullet]_t}\tilde{\mathbf{x}}^{[\bullet]_{t-1}}\right)' \hat{\pi}^{[\bullet]_t} + \sum_{l_t=1}^{L_t} \sum_{k_t=1}^{K_1^{[\bullet]_{t+1}}} b^{[\bullet]_{t+1}}_{1k_t} \hat{\eta}^{[\bullet]_{t+1}}_{k_t} \\ + \sum_{l_t=1}^{L_t} \sum_{k_t=1}^{K_2^{[\bullet]_{t+1}}} b^{[\bullet]_{t+1}}_{2k_t} \hat{v}^{[\bullet]_{t+1}}_{k_t} \end{array} \right].$$

Since the optimal dual solutions above are feasible but not necessarily optimum to problem **DWD RMP-SUB**($[\bullet]_t$) for any $\mathbf{x}^{[\bullet]_{t-1}}$, weak duality ensures that

$$\mathfrak{L}^{[\bullet]_t}\left(\mathbf{x}^{[\bullet]_{t-1}}\right) \leq p_{l_{t-1}}^{[\bullet]_{t-1}} \left[\begin{array}{r} \left(\mathbf{b}^{[\bullet]_t} - \mathbf{B}^{[\bullet]_t}\mathbf{x}^{[\bullet]_{t-1}}\right)' \hat{\boldsymbol{\pi}}^{[\bullet]_t} + \sum_{l_t=1}^{L_t} \sum_{k_t=1}^{K_1^{[\bullet]_{t+1}}} b_{1k_t}^{[\bullet]_{t+1}} \hat{\eta}_{k_t}^{[\bullet]_{t+1}} \\ + \sum_{l_t=1}^{L_t} \sum_{k_t=1}^{K_2^{[\bullet]_{t+1}}} b_{2k_t}^{[\bullet]_{t+1}} \hat{\upsilon}_{k_t}^{[\bullet]_{t+1}} \end{array} \right].$$

Set $K_1^{[\bullet]_t} = K_1^{[\bullet]_t} + 1$, let $k = K_1^{[\bullet]_t}$ and let

$$\begin{aligned} \boldsymbol{\pi}_k^{[\bullet]_t} &= p_{l_{t-1}}^{[\bullet]_{t-1}} \hat{\boldsymbol{\pi}}^{[\bullet]_t}, \\ \eta_{k_t}^{[\bullet]_{t+1}} &= p_{l_{t-1}}^{[\bullet]_{t-1}} \hat{\eta}_{k_t}^{[\bullet]_{t+1}}, \quad k_t = 1, \ldots, K_1^{[\bullet]_{t+1}}, \quad l_t = 1, \ldots, L_t, \\ \upsilon_{k_t}^{[\bullet]_{t+1}} &= p_{l_{t-1}}^{[\bullet]_{t-1}} \hat{\upsilon}_{k_t}^{[\bullet]_{t+1}}, \quad k_t = 1, \ldots, K_2^{[\bullet]_{t+1}}, \quad l_t = 1, \ldots, L_t. \end{aligned}$$

Then $\theta^{[\bullet]_t} \leq \mathfrak{Q}^{[\bullet]_t}\left(\mathbf{x}^{[\bullet]_{t-1}}\right) \leq \mathfrak{L}^{[\bullet]_t}\left(\mathbf{x}^{[\bullet]_{t-1}}\right)$ implies the optimality cut

$$\left(\boldsymbol{\pi}_k^{\prime[\bullet]_t}\mathbf{B}^{[\bullet]_t}\right)\mathbf{x}^{[\bullet]_{t-1}} + \theta^{[\bullet]_t} \leq \mathbf{b}^{\prime[\bullet]_t}\boldsymbol{\pi}_k^{[\bullet]_t} + \sum_{l_t=1}^{L_t} \sum_{k_t=1}^{K_1^{[\bullet]_{t+1}}} b_{1k_t}^{[\bullet]_{t+1}} \eta_{k_t}^{[\bullet]_{t+1}} + \sum_{l_t=1}^{L_t} \sum_{k_t=1}^{K_2^{[\bullet]_{t+1}}} b_{2k_t}^{[\bullet]_{t+1}} \upsilon_{k_t}^{[\bullet]_{t+1}}.$$

The right-hand-side of the above cut is the right-hand-side of the new optimality cut at the parent node in period $t-1$ for problem **LSD RMP-SUB**($[\bullet]_{t-1}$) [4.2] if $3 \leq t \leq T-1$ or problem **LSD RMP**($[\,]$) [4.3] if $t = 2$:

$$b_{1k}^{[\bullet]_t} = \mathbf{b}^{\prime[\bullet]_t}\boldsymbol{\pi}_k^{[\bullet]_t} + \sum_{l_t=1}^{L_t} \sum_{k_t=1}^{K_1^{[\bullet]_{t+1}}} b_{1k_t}^{[\bullet]_{t+1}} \eta_{k_t}^{[\bullet]_{t+1}} + \sum_{l_t=1}^{L_t} \sum_{k_t=1}^{K_2^{[\bullet]_{t+1}}} b_{2k_t}^{[\bullet]_{t+1}} \upsilon_{k_t}^{[\bullet]_{t+1}}. \qquad [4.8]$$

The coefficients for the new extreme point activity for the corresponding DWD problem are then

$$\left[\begin{array}{c} b_{1k}^{[\bullet]_t} \\ \mathbf{B}^{\prime[\bullet]_t}\boldsymbol{\pi}_k^{[\bullet]_t} \\ 1 \end{array} \right],$$

where the first element is the objective function coefficient for the new activity and the remaining elements are the technology matrix coefficients for the new activity.

Equation [4.8] indicates that the right-hand-side for an optimality cut at some period t node is a recursive function of the right-hand-sides of all optimality and feasibility cuts for all descendent nodes in periods $t+1, \ldots, T-1$. Therefore, for example, an optimality cut added to the single first period node for some outcome, say l_1, includes information for all descendent nodes with a first outcome of l_1 due to the nested sums in equation [4.8].

The terminal sum in recursive equation [4.8] incorporates the right-hand-sides, $b_{1k}^{[\bullet]_T}$, of the optimality cuts at nodes in period $T-1$. These values are created when problem **LSD SUB**$([\bullet]_T)$ [4.4] at a child node in the terminal period is solved to optimality for $\mathbf{x}^{[\bullet]_{T-1}} = \tilde{\mathbf{x}}^{[\bullet]_{T-1}}$. Let $\hat{\boldsymbol{\pi}}^{[\bullet]_T}$ be the optimal solutions to the corresponding dual problem **DWD SUB**$([\bullet]_T)$ [4.7], then strong duality implies that

$$\mathfrak{L}^{[\bullet]_T}\left(\tilde{\mathbf{x}}^{[\bullet]_{T-1}}\right) = p_{l_{T-1}}^{[\bullet]_{T-1}}\left(\mathbf{b}^{[\bullet]_T} - \mathbf{B}^{[\bullet]_T}\tilde{\mathbf{x}}^{[\bullet]_{T-1}}\right)'\hat{\boldsymbol{\pi}}^{[\bullet]_T}.$$

Since $\hat{\boldsymbol{\pi}}^{[\bullet]_T}$ are feasible but not necessarily optimum to problem **DWD SUB**$([\bullet]_T)$ for any $\mathbf{x}^{[\bullet]_{T-1}}$, weak duality ensures that

$$\mathfrak{L}^{[\bullet]_T}\left(\mathbf{x}^{[\bullet]_{T-1}}\right) \leq p_{l_{T-1}}^{[\bullet]_{T-1}}\left(\mathbf{b}^{[\bullet]_T} - \mathbf{B}^{[\bullet]_T}\mathbf{x}^{[\bullet]_{T-1}}\right)'\hat{\boldsymbol{\pi}}^{[\bullet]_T}.$$

Set $K_1^{[\bullet]_T} = K_1^{[\bullet]_T}+1$, let $k = K_1^{[\bullet]_T}$ and let $\boldsymbol{\pi}_k^{[\bullet]_T} = p_{l_{T-1}}^{[\bullet]_{T-1}}\hat{\boldsymbol{\pi}}^{[\bullet]_T}$, then $\theta^{[\bullet]_T} \leq \mathfrak{Q}^{[\bullet]_T}\left(\mathbf{x}^{[\bullet]_{T-1}}\right) \leq \mathfrak{L}^{[\bullet]_T}\left(\mathbf{x}^{[\bullet]_{T-1}}\right)$ implies the optimality cut

$$\left(\boldsymbol{\pi}_k'^{[\bullet]_T}\mathbf{B}^{[\bullet]_T}\right)\mathbf{x}^{[\bullet]_{T-1}} + \theta^{[\bullet]_T} \leq \mathbf{b}'^{[\bullet]_T}\boldsymbol{\pi}_k^{[\bullet]_T}.$$

The coefficients for the new extreme point activity for the corresponding DWD problem are then

$$\begin{bmatrix} \mathrm{b}'^{[\bullet]_T}\pi_k^{[\bullet]_T} \\ \mathrm{B}'^{[\bullet]_T}\pi_k^{[\bullet]_T} \\ 1 \end{bmatrix}.$$

The right-hand-side of the last inequality above coupled with equation [4.8] results in the recursive relationship:

$$b^{[\bullet]_t}_{1\left(K_1^{[\bullet]_t}+1\right)} = \begin{cases} \mathrm{b}'^{[\bullet]_t}\pi^{[\bullet]_t}_{K_1^{[\bullet]_t}+1}, & \text{if } t = T, \\ \mathrm{b}'^{[\bullet]_t}\pi^{[\bullet]_t}_{K_1^{[\bullet]_t}+1} + \sum_{l_t=1}^{L_t}\sum_{k=1}^{K_1^{[\bullet]_{t+1}}} b^{[\bullet]_{t+1}}_{1k}\eta_k^{[\bullet]_{t+1}} & \\ \qquad + \sum_{l_t=1}^{L_t}\sum_{k=1}^{K_2^{[\bullet]_{t+1}}} b^{[\bullet]_{t+1}}_{2k}v_k^{[\bullet]_{t+1}}, & \text{if } 2 \le t \le T-1, \end{cases} \qquad [4.9]$$

for the right-hand-sides of optimality cuts in LSD or the objective function coefficients of extreme point activities in DWD. Note that the derived cuts/activities are based upon on a bounded solution to the parent relaxed master problem for LSD or a feasible solution to the parent restricted master problem for DWD. Procedures for an unbounded/infeasible RMP are straightforward extensions to the corresponding methods developed in Sections 3.1.3 and 3.2.2.

4.2.3 Feasibility Cuts – Extreme Direction Activities

Derivations are given for the feasibility cuts of L-Shaped decomposition of the primal problem with the corresponding translation to extreme direction activities for Dantzig-Wolfe decomposition of the dual problem. Recall that the RMP is augmented with a new feasibility cut whenever a subproblem is determined to be infeasible. Assume that problem **LSD**

RMP-SUB$([\bullet]_t)$ [4.2] is infeasible with $\mathbf{x}^{[\bullet]_{t-1}} = \tilde{\mathbf{x}}^{[\bullet]_{t-1}}$ at a node in an intermediate period, $2 \leq t \leq T-1$, at some LSD iteration. Then the vector of coefficients, $\boldsymbol{\delta}^{[\bullet]_t}_{K_2^{[\bullet]_t}+1}$, and the right-hand-side, $b^{[\bullet]_t}_{2\left(K_2^{[\bullet]_t}+1\right)}$, for the new feasibility cut for the RMP at the parent node in period $t-1$ must be determined. Based upon problem **LSD RMP-SUB**$([\bullet]_t)$, solve the following simplex phase one type problem with $\mathbf{x}^{[\bullet]_{t-1}} = \tilde{\mathbf{x}}^{[\bullet]_{t-1}}$:

$$
\begin{aligned}
\mathcal{W}^{[\bullet]_t}\left(\mathbf{x}^{[\bullet]_{t-1}}\right) = \min \quad & \mathbf{v} + \sum_{l_t=1}^{L_t} \sum_{k=1}^{K_1^{[\bullet]_{t+1}}} w_{1k}^{(l_t)} + \sum_{l_t=1}^{L_t} \sum_{k=1}^{K_2^{[\bullet]_{t+1}}} w_{2k}^{(l_t)} \\
\text{s.t.} \quad & \mathbf{I}\mathbf{v} - \mathbf{W}^{[\bullet]_t}\mathbf{x}^{[\bullet]_t} \geq \mathbf{B}^{[\bullet]_t}\mathbf{x}^{[\bullet]_{t-1}} - \mathbf{b}^{[\bullet]_t}, \\
& w_{1k}^{(l_t)} - \boldsymbol{\pi}_k^{\prime[\bullet]_{t+1}}\mathbf{B}^{[\bullet]_{t+1}}\mathbf{x}^{[\bullet]_t} + \theta^{[\bullet]_{t+1}} \geq -b_{1k}^{[\bullet]_{t+1}}, \quad \begin{matrix} k = 1, \ldots, K_1^{[\bullet]_{t+1}}, \\ l_t = 1, \ldots, L_t, \end{matrix} \\
& w_{2k}^{(l_t)} - \boldsymbol{\delta}_k^{\prime[\bullet]_{t+1}}\mathbf{B}^{[\bullet]_{t+1}}\mathbf{x}^{[\bullet]_t} \geq -b_{2k}^{[\bullet]_{t+1}}, \quad \begin{matrix} k = 1, \ldots, K_2^{[\bullet]_{t+1}}, \\ l_t = 1, \ldots, L_t, \end{matrix} \\
& \mathbf{x}^{[\bullet]_t} \geq \mathbf{0}, \\
& \theta^{[\bullet]_{t+1}} \text{ free}, \; l_t = 1, \ldots, L_t, \\
& \mathbf{v} \geq \mathbf{0}, \\
& w_{1k}^{(l_t)} \geq 0, \quad \begin{matrix} k = 1, \ldots, K^{[\bullet]_{t+1}}, \\ l_t = 1, \ldots, L_t, \end{matrix} \\
& w_{2k}^{(l_t)} \geq 0, \quad \begin{matrix} k = 1, \ldots, K^{[\bullet]_{t+1}}, \\ l_t = 1, \ldots, L_t, \end{matrix}
\end{aligned}
\qquad [4.10]
$$

which is always feasible and bounded and corresponds to the dual problem:

$$\mathcal{W}^{[\bullet]_t}\left(\mathbf{x}^{[\bullet]_{t-1}}\right) = \max \quad \left(\mathbf{B}^{[\bullet]_t}\mathbf{x}^{[\bullet]_{t-1}} - \mathbf{b}^{[\bullet]_t}\right)' \boldsymbol{\delta}^{[\bullet]_t} - \sum_{l_t=1}^{L_t} \sum_{k=1}^{K_1^{[\bullet]_{t+1}}} b_{1k}^{[\bullet]_{t+1}} \tau_k^{[\bullet]_{t+1}} - \sum_{l_t=1}^{L_t} \sum_{k=1}^{K_2^{[\bullet]_{t+1}}} b_{2k}^{[\bullet]_{t+1}} \rho_k^{[\bullet]_{t+1}}$$

$$\text{s.t.} \quad \mathbf{W}'^{[\bullet]_t}\boldsymbol{\delta}^{[\bullet]_t} + \sum_{l_t=1}^{L_t} \sum_{k=1}^{K_1^{[\bullet]_{t+1}}} \mathbf{B}'^{[\bullet]_{t+1}} \boldsymbol{\pi}_k^{[\bullet]_{t+1}} \tau_k^{[\bullet]_{t+1}} + \sum_{l_t=1}^{L_t} \sum_{k=1}^{K_2^{[\bullet]_{t+1}}} \mathbf{B}'^{[\bullet]_{t+1}} \boldsymbol{\delta}_k^{[\bullet]_{t+1}} \rho_k^{[\bullet]_{t+1}} \geq \mathbf{0}, \qquad [4.11]$$

$$\sum_{k=1}^{K_1^{[\bullet]_{t+1}}} \tau_k^{[\bullet]_{t+1}} = 0, \; l_t = 1, \ldots, L_t,$$

$$\boldsymbol{\delta}^{[\bullet]_t} \geq \mathbf{0},$$

$$\tau_k^{[\bullet]_{t+1}} \geq 0, \quad \begin{matrix} k = 1, \ldots, K_1^{[\bullet]_{t+1}}, \\ l_t = 1, \ldots, L_t, \end{matrix}$$

$$\rho_k^{[\bullet]_{t+1}} \geq 0, \quad \begin{matrix} k = 1, \ldots, K_2^{[\bullet]_{t+1}}, \\ l_t = 1, \ldots, L_t. \end{matrix}$$

Set $K_2^{[\bullet]_t} = K_2^{[\bullet]_t} + 1$, let $k = K_2^{[\bullet]_t}$ and let the optimal solution to the dual problem [4.11] be

$$\begin{matrix} \boldsymbol{\delta}_k^{[\bullet]_t}, & & & \\ \tau_{k_t}^{[\bullet]_{t+1}} = 0, & k_t = 1, \ldots, K_1^{[\bullet]_{t+1}}, & l_t = 1, \ldots, L_t, \\ \rho_{k_t}^{[\bullet]_{t+1}}, & k_t = 1, \ldots, K_2^{[\bullet]_{t+1}}, & l_t = 1, \ldots, L_t, \end{matrix}$$

then

$$\mathcal{W}^{[\bullet]_t}\left(\tilde{\mathbf{x}}^{[\bullet]_{t-1}}\right) = \left(\mathbf{B}^{[\bullet]_t}\tilde{\mathbf{x}}^{[\bullet]_{t-1}} - \mathbf{b}^{[\bullet]_t}\right)' \boldsymbol{\delta}_k^{[\bullet]_t} - \sum_{l_t=1}^{L_t} \sum_{k_t=1}^{K_2^{[\bullet]_{t+1}}} b_{2k_t}^{[\bullet]_{t+1}} \rho_{k_t}^{[\bullet]_{t+1}} > 0$$

since the original problem **LSD RMP-SUB**($[\bullet]_t$) [4.2] is infeasible. Therefore, the feasibility cut

$$\left(\delta_k'^{[\bullet]_t}\mathrm{B}^{[\bullet]_t}\right)\mathrm{x}^{[\bullet]_{t-1}} \leq \mathrm{b}'^{[\bullet]_t}\delta_k^{[\bullet]_t} + \sum_{l_t=1}^{L_t}\sum_{k_t=1}^{K_2^{[\bullet]_{t+1}}} b_{2k_t}^{[\bullet]_{t+1}}\rho_{k_t}^{[\bullet]_{t+1}}$$

will not admit the infeasible point $\tilde{\mathrm{x}}^{[\bullet]_{t-1}}$. Furthermore, $\mathcal{W}^{[\bullet]_t}\left(\mathrm{x}^{[\bullet]_{t-1}}\right) = 0$ for any $\mathrm{x}^{[\bullet]_{t-1}}$ feasible to problem **LSD RMP-SUB**($[\bullet]_t$) implying that

$$\mathcal{W}^{[\bullet]_t}\left(\mathrm{x}^{[\bullet]_{t-1}}\right) = 0 \geq \left(\mathrm{B}^{[\bullet]_t}\mathrm{x}^{[\bullet]_{t-1}} - \mathrm{b}^{[\bullet]_t}\right)'\delta_k^{[\bullet]_t} - \sum_{l_t=1}^{L_t}\sum_{k_t=1}^{K_2^{[\bullet]_{t+1}}} b_{2k_t}^{[\bullet]_{t+1}}\rho_{k_t}^{[\bullet]_{t+1}}$$

for any $\mathrm{x}^{[\bullet]_{t-1}}$ feasible to problem **LSD RMP-SUB**($[\bullet]_t$) since the above solutions are feasible but not necessarily optimum to the dual simplex phase one type problem [4.11]. The above cut will therefore not exclude any $\mathrm{x}^{[\bullet]_{t-1}}$ feasible to problem **LSD RMP-SUB**($[\bullet]_t$). The right-hand-side of the above cut is the right-hand-side of the new feasibility cut for problem **LSD RMP-SUB**($[\bullet]_{t-1}$) [4.2] if $3 \leq t \leq T-1$ or problem **LSD RMP**($[\,]$) [4.3] if $t = 2$ at the parent node in period $t-1$:

$$b_{2k}^{[\bullet]_t} = \mathrm{b}'^{[\bullet]_t}\delta_k^{[\bullet]_t} + \sum_{l_t=1}^{L_t}\sum_{k_t=1}^{K_2^{[\bullet]_{t+1}}} b_{2k_t}^{[\bullet]_{t+1}}\rho_{k_t}^{[\bullet]_{t+1}}. \qquad [4.12]$$

The coefficients for the new extreme direction activity for the corresponding DWD problem are then

$$\begin{bmatrix} b_{2k}^{[\bullet]_t} \\ \mathrm{B}'^{[\bullet]_t}\delta_k^{[\bullet]_t} \\ 0 \end{bmatrix},$$

where the first element is the objective function coefficient for the new activity and the remaining elements are the technology matrix coefficients for the new activity.

Equation [4.12] indicates that the right-hand-side for an feasibility cut at some period t node is a recursive function of the right-hand-sides of all feasibility cuts for all descendent nodes in periods $t+1,\ldots,T-1$. Therefore, for example, a feasibility cut added to the single first period node for some outcome, say l_1, includes information for all descendent nodes with a first outcome of l_1 due to the nested sums in equation [4.12].

The terminal sum in recursive equation [4.12] incorporates the right-hand-sides, $b_{2k}^{[\bullet]_T}$, of the feasibility cuts at nodes in period $T-1$. These values are created when problem **LSD SUB**$([\bullet]_T)$ [4.4] at a child node in the terminal period is infeasible for $\mathbf{x}^{[\bullet]_{T-1}} = \tilde{\mathbf{x}}^{[\bullet]_{T-1}}$. Based upon problem **LSD SUB**$([\bullet]_T)$, solve the following simplex phase one type problem with $\mathbf{x}^{[\bullet]_{T-1}} = \tilde{\mathbf{x}}^{[\bullet]_{T-1}}$:

$$\begin{aligned} \mathcal{W}^{[\bullet]_T}\left(\mathbf{x}^{[\bullet]_{T-1}}\right) = \min \quad & \mathbf{v} & & \\ \text{s.t.} \quad & \mathbf{I}\mathbf{v} - \mathbf{W}^{[\bullet]_T}\mathbf{x}^{[\bullet]_T} & \geq & \mathbf{B}^{[\bullet]_T}\mathbf{x}^{[\bullet]_{T-1}} - \mathbf{b}^{[\bullet]_T}, \\ & \mathbf{x}^{[\bullet]_T} & \geq & \mathbf{0}, \\ & \mathbf{v} & \geq & \mathbf{0}, \end{aligned} \qquad [4.13]$$

which is always feasible and bounded and corresponds to the dual problem:

$$\begin{aligned} \mathcal{W}^{[\bullet]_T}\left(\mathbf{x}^{[\bullet]_{T-1}}\right) = \max \quad & \left(\mathbf{B}^{[\bullet]_T}\mathbf{x}^{[\bullet]_{T-1}} - \mathbf{b}^{[\bullet]_T}\right)'\boldsymbol{\delta}^{[\bullet]_T} & & \\ \text{s.t.} \quad & \mathbf{W}'^{[\bullet]_T}\boldsymbol{\delta}^{[\bullet]_T} & \geq & \mathbf{0}, \\ & \boldsymbol{\delta}^{[\bullet]_T} & \leq & \mathbf{1}, \\ & \boldsymbol{\delta}^{[\bullet]_T} & \geq & \mathbf{0}. \end{aligned} \qquad [4.14]$$

Set $K_2^{[\bullet]_T} = K_2^{[\bullet]_T} + 1$, let $k = K_2^{[\bullet]_T}$ and let the optimal solution to the dual problem [4.14] be $\boldsymbol{\delta}_k^{[\bullet]_t}$, then

$$\mathcal{W}^{[\bullet]^T}\left(\tilde{\mathbf{x}}^{[\bullet]_{t-1}}\right) = \left(\mathbf{B}^{[\bullet]_T}\tilde{\mathbf{x}}^{[\bullet]_{T-1}} - \mathbf{b}^{[\bullet]_T}\right)'\boldsymbol{\delta}_k^{[\bullet]_T} > 0$$

since the original problem **LSD SUB**$([\bullet]_T)$ [4.4] is infeasible. Therefore, the feasibility cut

$$\left(\boldsymbol{\delta}_k'^{[\bullet]_T}\mathbf{B}^{[\bullet]_T}\right)\mathbf{x}^{[\bullet]_{T-1}} \leq \mathbf{b}'^{[\bullet]_T}\boldsymbol{\delta}_k^{[\bullet]_T}$$

will not admit the infeasible point $\tilde{\mathbf{x}}^{[\bullet]_{T-1}}$. Furthermore, $\mathcal{W}^{[\bullet]_T}\left(\mathbf{x}^{[\bullet]_{T-1}}\right) = 0$ for any $\mathbf{x}^{[\bullet]_{T-1}}$ feasible to problem **LSD SUB**$([\bullet]_T)$ implying that

$$\mathcal{W}^{[\bullet]_T}\left(\mathbf{x}^{[\bullet]_{T-1}}\right) = 0 \geq \left(\mathbf{B}^{[\bullet]_T}\mathbf{x}^{[\bullet]_{T-1}} - \mathbf{b}^{[\bullet]_T}\right)' \boldsymbol{\delta}_k^{[\bullet]_T}$$

for any $\mathbf{x}^{[\bullet]_{T-1}}$ feasible to problem **LSD SUB**$([\bullet]_T)$ since the above solutions are feasible but not necessarily optimum to the dual simplex phase one type problem [4.14]. The above cut will therefore not exclude any $\mathbf{x}^{[\bullet]_{T-1}}$ feasible to problem **LSD SUB**$([\bullet]_T)$. The coefficients for the new extreme direction activity for the corresponding DWD problem are then

$$\begin{bmatrix} \mathbf{b}'^{[\bullet]_T}\boldsymbol{\delta}_k^{[\bullet]_T} \\ \mathbf{B}'^{[\bullet]_T}\boldsymbol{\delta}_k^{[\bullet]_T} \\ 0 \end{bmatrix}.$$

The right-hand-side of the last cut above coupled with equation [4.12] results in the recursive relationship:

$$b_{2\left(K_2^{[\bullet]_t}+1\right)}^{[\bullet]_t} = \begin{cases} \mathbf{b}'^{[\bullet]_t}\boldsymbol{\delta}_{K_2^{[\bullet]_t}+1}^{[\bullet]_t}, & \text{if } t = T, \\ \mathbf{b}'^{[\bullet]_t}\boldsymbol{\delta}_{K_2^{[\bullet]_t}+1}^{[\bullet]_t} + \sum_{l_t=1}^{L_t}\sum_{k=1}^{K_2^{[\bullet]_{t+1}}} b_{2k}^{[\bullet]_{t+1}}\rho_k^{[\bullet]_{t+1}}, & \text{if } 2 \leq t \leq T-1, \end{cases} \qquad [4.15]$$

for the right-hand-sides of feasibility cuts in LSD or the objective function coefficients of extreme direction activities in DWD. Note that the derived cuts/activities are based upon on a bounded solution to the parent relaxed master problem for LSD or a feasible solution to the parent restricted master problem for DWD. Procedures for an unbounded/infeasible RMP are straightforward extensions to the corresponding methods developed in Sections 3.1.3 and 3.2.2.

4.2.4 Nested Decomposition Algorithms

The nested decomposition algorithms cited at the beginning of this section are similar to the multicut algorithms developed in Chapter 3 – each incorporates distinct initialization and optimization procedures. Initialization procedures determine if the problem is unbounded or infeasible, or, otherwise, find an initial feasible solution to the problem. Optimization procedures then determine an ε-optimal solution given the problem is feasible and bounded. The procedures are summarized below using L-Shaped decomposition with straightforward translations for Dantzig-Wolfe decomposition.

Initialization Procedure

The initialization procedure consists of a forward pass followed by a backward pass through all nodes in the decision tree. The forward pass begins by finding a solution, $\mathbf{x}^{[]}$, to problem **LSD RMP**([ ]) [4.3] at the first period node. The overall problem is infeasible if **LSD RMP**([ ]) is infeasible, otherwise $\mathbf{x}^{[]}$ is passed to problems **LSD RMP-SUB**$([\bullet]_2)$ [4.2] at each second period node. A feasibility cut is generated for **LSD RMP**([ ]) for each **LSD RMP-SUB**$([\bullet]_2)$ that is infeasible and **LSD RMP**([ ]) is resolved a for new solution. Note that a single feasibility cut may also be passed back if control is returned to **LSD RMP**([ ]) when the first infeasible **LSD RMP-SUB**$([\bullet]_2)$ is encountered. This procedure is then repeated until all second period subproblems are feasible or the overall problem is determined to be infeasible.

The steps above are repeated at each node $[\bullet]_t$ in the decision tree for $t = 2, \ldots, T$. Problem **LSD RMP-SUB**($[\bullet]_t$), $t = 2, \ldots, T-1$, receives a solution vector $\mathbf{x}^{[\bullet]_{t-1}}$ from its parent node and either returns a feasibility cut to the parent or passes a solution vector $\mathbf{x}^{[\bullet]_t}$ to each of its child nodes. Solution vectors are passed forward only when the problems at all child nodes with a common parent are feasible. This process continues until the problem is determined to be infeasible or feasible solutions are found to problems **LSD SUB**($[\bullet]_T$) [4.4] at all nodes in the terminal period. The overall problem is unbounded if a feasible solution is determined at all nodes and an ascent direction is found at any node in the tree with no corresponding cut to exclude the resulting ray or admit only a portion of the ray. A lower bound on the objective function of the overall problem is then generated given a feasible and bounded solution at all nodes in the tree.

A backward pass is executed only if a feasible and bounded solution is found during the forward pass through the tree. The purpose of this pass is two fold: determine an initial upper bound on the overall problem; and place bounds on each $\theta^{[\bullet]_{t+1}}$ at all nodes in periods $t = 2, \ldots, T-1$ and generate initial optimality cuts at the first period node. Given a feasible solution, $\tilde{\mathbf{x}}^{[\bullet]_T}$, at a terminal period node $[\bullet]_T$, the corresponding $\theta^{[\bullet]_T}$ at the parent node in period $T-1$ is bounded by:

$$\theta^{[\bullet]_T} \leq \Omega^{[\bullet]_T}\left(\tilde{\mathbf{x}}^{[\bullet]_{T-1}}\right) = p_{l_{T-1}}^{[\bullet]_{T-1}} \mathbf{c}'^{[\bullet]_T} \tilde{\mathbf{x}}^{[\bullet]_T}.$$

This bound is *looser* than the corresponding cut constraint in the sense that it bounds only $\theta^{[\bullet]_T}$ without placing restrictions on $\mathbf{x}^{[\bullet]_{T-1}}$. Bounds are used in lieu of the normal cut

constraints since, otherwise, problem **LSD RMP-SUB**($[\bullet]_t$), $t = 2, \ldots, T-1$, would need to be reinitialized with each new $\tilde{\mathbf{x}}^{[\bullet]_{t-1}}$. Cut constraints at a node $[\bullet]_t$ in an intermediate period, $2 \leq t \leq T-1$, are dependent upon the feasible solution vector $\tilde{\mathbf{x}}^{[\bullet]_{t-1}}$ received from the parent node. Maintaining cuts generated for one solution, say $\tilde{\mathbf{x}}_1^{[\bullet]_{t-1}}$, with a new solution, $\tilde{\mathbf{x}}_2^{[\bullet]_{t-1}}$, would invalidate the procedure. This would be akin to generating cuts for the first period node with one set of values for fixed $\mathbf{b}^{[]}$ and keeping these cuts after creating an entirely new problem by changing the right-hand-side values.

The backward pass then continues through the nodes in periods $t = T-1, \ldots, 3$ by placing bounds on each $\theta^{[\bullet]_t}$ with the recursive relation:

$$\theta^{[\bullet]_t} \leq \mathfrak{Q}^{[\bullet]_t}\left(\tilde{\mathbf{x}}^{[\bullet]_{t-1}}\right) = p_{l_{t-1}}^{[\bullet]_{t-1}} \left[\mathbf{c}'^{[\bullet]_t} \tilde{\mathbf{x}}^{[\bullet]_t} + \sum_{l_t=1}^{L_t} \mathfrak{Q}^{[\bullet]_{t+1}}\left(\tilde{\mathbf{x}}^{[\bullet]_t}\right) \right].$$

This pass and the initialization procedure terminates when the normal optimality cuts are generated for the first period node by each second period node. An initial upper bound on the objective function of the overall problem is then established as:

$$z^{\mathrm{P}} \leq \mathbf{c}'^{[]} \tilde{\mathbf{x}}^{[]} + \sum_{l_1=1}^{L_1} \mathfrak{Q}^{[\bullet]_2}\left(\tilde{\mathbf{x}}^{[]}\right).$$

Optimization Procedure

The optimization procedure is executed only if the initialization procedure determines that the overall problem is feasible and bounded. Processing starts by finding a solution to problem **LSD RMP**([ ]) [4.3] at the first period node and then moving through the tree by passing feasible solutions forward or cuts backward. A new upper bound is created each

time problem **LSD RMP**([ ]) is solved (assuming nondegenerency). The lower bound is updated whenever feasible solutions are found at all nodes in the terminal period implying a feasible solution to the overall problem. Processing terminates whenever the relative difference between the upper and lower bounds is less than some specified positive value ε.

All existing cuts, except for the upper bounds on $\theta^{[\bullet]_{t+1}}$, at problem **LSD RMP-SUB**($[\bullet]_t$), $t = 2, \ldots, T-1$, must be removed each time the parent node sends a new solution vector. Existing cuts are removed with a new parent solution vector for the reasons discussed above with the backward pass of the initialization procedure.

Several *sequencing protocols* (Birge et al. [14, Section 2.2] and Gassmann [83, page 414]) for controlling the traversal through the tree are discussed in the literature. A few of these protocols are described in Section 4.4.2 below.

4.3 Block-Separable Problems

Stochastic linear programs that have the block-separable property defined in Section 2.4 provide much greater flexibility in the application of decomposition algorithms than do programs without this property. Given block-separability, the multi-period primal stochastic linear programming problem [2.8] on page 35 may be written in the equivalent form:

$$z^{\mathrm{P}} = \max \ \mathbf{c}'^{(1)[]}\mathbf{x}^{(1)[]} + \mathbf{c}'^{(2)[]}\mathbf{x}^{(2)[]} + \sum_{l_1=1}^{L_1} p_{l_1}^{[]} \left[\mathbf{c}'^{(1)[\bullet]_2}\mathbf{x}^{(1)[\bullet]_2} + \mathbf{c}'^{(1)[\bullet]_2}\mathbf{x}^{(1)[\bullet]_2}\right.$$
$$+ \sum_{l_2=1}^{L_2} p_{l_2}^{[\bullet]_2} \left[\mathbf{c}'^{(1)[\bullet]_3}\mathbf{x}^{(1)[\bullet]_3} + \mathbf{c}'^{(2)[\bullet]_3}\mathbf{x}^{(2)[\bullet]_3}\right.$$
$$\left.\left. + \ldots + \sum_{l_{T-1}=1}^{L_{T-1}} p_{l_{T-1}}^{[\bullet]_{T-1}} \left[\mathbf{c}'^{(1)[\bullet]_T}\mathbf{x}^{(1)[\bullet]_T} + \mathbf{c}'^{(1)[\bullet]_T}\mathbf{x}^{(1)[\bullet]_T} \ldots\right]\right]\right]$$

s.t.

$$\begin{aligned}
\mathbf{A}^{(1,1)}\mathbf{x}^{(1)[]} &\leq \mathbf{b}^{(1)[]}, \\
\mathbf{A}^{(2,1)}\mathbf{x}^{(1)[]} + \mathbf{A}^{(2,2)}\mathbf{x}^{(2)[]} &\leq \mathbf{b}^{(2)[]}, \\
\mathbf{B}^{(1,1)[\bullet]_t}\mathbf{x}^{(1)[\bullet]_{t-1}} + \mathbf{W}^{(1,1)[\bullet]_t}\mathbf{x}^{(1)[\bullet]_t} &\leq \mathbf{b}^{(1)[\bullet]_t}, \left\{ \begin{array}{l} h_t = 1, \ldots, H_t, \\ t = 2, \ldots, T, \end{array} \right. \\
\mathbf{B}^{(2,1)[\bullet]_t}\mathbf{x}^{(1)[\bullet]_{t-1}} + \mathbf{W}^{(2,1)[\bullet]_t}\mathbf{x}^{(1)[\bullet]_t} + \mathbf{W}^{(2,2)[\bullet]_t}\mathbf{x}^{(2)[\bullet]_t} &\leq \mathbf{b}^{(2)[\bullet]_t}, \left\{ \begin{array}{l} h_t = 1, \ldots, H_t, \\ t = 2, \ldots, T, \end{array} \right.
\end{aligned}$$

$$\begin{aligned}
\mathbf{x}^{(1)[]}, \mathbf{x}^{(2)[]} &\geq \mathbf{0}, \\
\mathbf{x}^{(1)[\bullet]_t}, \mathbf{x}^{(2)[\bullet]_t} &\geq \mathbf{0}, \left\{ \begin{array}{l} h_t = 1, \ldots, H_t, \\ t = 2, \ldots, T, \end{array} \right.
\end{aligned} \qquad [4.16]$$

where equation [2.7] on page 28 is used to determine $[\bullet]_t$ given period-index node label (t, h_t) in the applicable constraints and in the discussion below. Then all constraints involving only the aggregate level variables, $\mathbf{x}^{(1)[\bullet]_t}$, can be grouped together as can the constraints that incorporate detailed level variables, $\mathbf{x}^{(2)[\bullet]_t}$. Problem [4.16] can then be rewritten as:

$$z^{\mathrm{P}} = \max \ \mathbf{c}'^{(1)[]}\mathbf{x}^{(1)[]} + \sum_{l_1=1}^{L_1} p_{l_1}^{[]} \left[\mathbf{c}'^{(1)[\bullet]_2}\mathbf{x}^{(1)[\bullet]_2} + \sum_{l_2=1}^{L_2} p_{l_2}^{[\bullet]_2} \left[\mathbf{c}'^{(1)[\bullet]_3}\mathbf{x}^{(1)[\bullet]_3} \right.\right.$$
$$\left.\left. + \sum_{l_{T-1}=1}^{L_{T-1}} p_{l_{T-1}}^{[\bullet]_{T-1}} \left[\mathbf{c}'^{(1)[\bullet]_T}\mathbf{x}^{(1)[\bullet]_T} \ldots\right]\right]\right]$$
$$\mathbf{c}'^{(2)[]}\mathbf{x}^{(2)[]} + \sum_{l_1=1}^{L_1} p_{l_1}^{[]} \left[\mathbf{c}'^{(2)[\bullet]_2}\mathbf{x}^{(2)[\bullet]_2} + \sum_{l_2=1}^{L_2} p_{l_2}^{[\bullet]_2} \left[\mathbf{c}'^{(2)[\bullet]_3}\mathbf{x}^{(2)[\bullet]_3} \right.\right.$$
$$\left.\left. + \sum_{l_{T-1}=1}^{L_{T-1}} p_{l_{T-1}}^{[\bullet]_{T-1}} \left[\mathbf{c}'^{(2)[\bullet]_T}\mathbf{x}^{(2)[\bullet]_T} \ldots\right]\right]\right]$$

s.t.

$$\begin{aligned}
\mathbf{A}^{(1,1)}\mathbf{x}^{(1)[]} &\leq \mathbf{b}^{(1)[]}, \\
\mathbf{B}^{(1,1)[\bullet]_t}\mathbf{x}^{(1)[\bullet]_{t-1}} + \mathbf{W}^{(1,1)[\bullet]_t}\mathbf{x}^{(1)[\bullet]_t} &\leq \mathbf{b}^{(1)[\bullet]_t}, \left\{ \begin{array}{r} h_t = 1, \ldots, H_t, \\ t = 2, \ldots, T, \end{array} \right. \\
\mathbf{A}^{(2,1)}\mathbf{x}^{(1)[]} + \mathbf{A}^{(2,2)}\mathbf{x}^{(2)[]} &\leq \mathbf{b}^{(2)[]}, \\
\mathbf{B}^{(2,1)[\bullet]_t}\mathbf{x}^{(1)[\bullet]_{t-1}} + \mathbf{W}^{(2,1)[\bullet]_t}\mathbf{x}^{(1)[\bullet]_t} + \mathbf{W}^{(2,2)[\bullet]_t}\mathbf{x}^{(2)[\bullet]_t} &\leq \mathbf{b}^{(2)[\bullet]_t}, \left\{ \begin{array}{r} h_t = 1, \ldots, H_t, \\ t = 2, \ldots, T, \end{array} \right. \\
\mathbf{x}^{(1)[]}, \mathbf{x}^{(2)[]} &\geq \mathbf{0}, \\
\mathbf{x}^{(1)[\bullet]_t}, \mathbf{x}^{(2)[\bullet]_t} &\geq \mathbf{0}, \left\{ \begin{array}{r} h_t = 1, \ldots, H_t, \\ t = 2, \ldots, T, \end{array} \right.
\end{aligned} \qquad ([4.17])$$

Then, with suitable definitions for composite arrays, the multi-period problem [4.17] can be rewritten to resemble a two-period problem. This means that the two-phase **LSD (multicut)** algorithm defined in Section 3.1.4 can be used in lieu of nested decomposition to solve a block-separable multi-period problem. Given a phase one solution for the aggregate level variables, the second pair of constraint sets in problem [4.17] are separable by node into phase two subproblems for the detailed level variables. The structures of matrices $\mathbf{A}^{(2,2)}$ and $\mathbf{W}^{(2,2)[\bullet]_t}$ may allow for even more separability so that the indicated constraints

are also separable into subproblems. This separability provides much greater flexibility than can be expected with nested decomposition. Similar remarks apply for using algorithm **DWD(multiactivities)** from Section 3.2.3 to obtain a solution to the dual formulation of a block-separable multi-period problem.

A disadvantage of bringing all constraints with only aggregate level variables into the first phase of the LSD procedure or bringing all aggregate level variables into the first phase of the DWD procedure is that the size of the corresponding initial RMP can become very large. Large initial relaxed/restricted master problems can require significant amounts of processing time in the early iterations of the decomposition algorithms. This can be a serious drawback since two of the generally expected advantages of decomposition are relative little memory usage and fast solution times in the early iterations. Note, however, that the formulation of problem [4.17] is just one of many ways of writing a block-separable multi-period problem to resemble a two-period problem. Constraints may be arranged by periods so that constraints involving only aggregate level variables at nodes in a specified number of early periods are brought into the first phase RMP while those in the later periods remain as second phase subproblem constraints. This latter arrangement provides for a smaller initial RMP at the expense of larger, more complicated, subproblems. A smaller initial RMP can still be advantageous since not all subproblems must be solved at every iteration – see the remarks on greedy algorithms in Section 3.3.6 on page 114. This and other issues are discussed in the next section on implementation strategies.

4.4 DWD-LSD Implementation Strategies

Several implementation strategies are possible when employing one of the decomposition methods to solve a multi-period stochastic linear program. The best strategy depends upon the system resources available as well as the structure of the problem. Determining the most efficient strategy for a given problem or class of problems may require extensive experimentation. Several implementation strategies are examined in this section. Empirical results for several of the strategies described below are given in Chapter 6.

Terminology used herein relative to that normally found in the literature is discussed in the first subsection below. The remaining three subsections consider implementation strategies for nested decomposition, two-phase decomposition of a block-separable problem, and the nested decomposition of a block-separable problem. Strategies are described relative to L-Shaped decomposition with simple translations possible for Dantzig-Wolfe decomposition.

4.4.1 Terminology Issues

The most common term used in the literature on stochastic linear programming is *stage*. Problems with a horizon of multiple (three or more) distinct periods are usually called *multi-stage* problems versus the term multi-period used in this thesis. L-Shaped and Dantzig-Wolfe decomposition are often referred to as *two-stage* procedures instead of two-phase procedures. Finally, the term *multi-stage decomposition* is frequently used synonymously with nested decomposition. The terms *periods* and *stages* were used interchangeably in the

early days of linear programming to describe the different parts of a program that would be separable except for a few linking constraints or variables, e.g., see Dantzig [42]. Stage has become the dominant term in stochastic programming. Overuse of the term can be confusing – does *stage* refer to a portion of the problem or to a part of the decomposition procedure. This is not really a problem in the references on nested decomposition cited at the beginning of Section 4.2. Those references employ the algorithm so that each stage of the procedure coincides with a stage or period of the problem. The flexibility provided by block-separable problems can, however, lead to obscure meanings – multi-stage decomposition could imply decomposition of a multi-stage problem with a two-stage algorithm. Therefore, *period* is used to refer to the distinguishable parts of a linear program in its original grand LP formulation and *phase* is used to differentiate between the two modes of a decomposition algorithm, i.e., the RMP phase versus the SUB phase. The term *stage* is used to identify a portion of the problem structure specifically constructed for a decomposition procedure. For instance, each stage may coincide with a period if nested decomposition is used as described above. On the other hand, a multi-period stochastic linear program with block-separable recourse may be restructured with two stages for decomposition.

4.4.2 Nested Decomposition Strategies

Several sequencing protocols are possible for controlling the order in which problems are solved in the optimization phase of a nested decomposition algorithm. Three of the most common protocols are described below. One or more of these protocols are discussed in

each of: Birge et al. [14, Section 2.2], Gassmann [83, pages 414-415], and Wittrock [207, pages 82-83]. The terminology is from Birge et al. and each description terminates with an enclosed listing of the contributing authors.

Fast Forward-Fast Back (FFFB): The algorithm initially begins with a forward pass starting with a solution to the first period node. Feasible solutions are passed forward through the tree as fast as possible returning to a parent node only when a corresponding subproblem generates a feasibility cut. A backward pass is initiated when all subproblems at nodes in the terminal period are solved and have generated, where applicable, either a feasibility cut or an optimality cut for their parent nodes. Cuts are passed back through the tree as fast as possible in the backward pass, i.e., one per child node (which may be aggregated in a single-cut algorithm). The backward pass terminates and a new forward pass is initiated whenever:

(a) all second period nodes have generated a cut for the first period, or

(b) no cut can be generated at some node $[\bullet]_t$, $t = 2, \ldots, T-1$, because problem **LSD RMP-SUB**$([\bullet]_t)$ [4.2] is optimum given feasible solutions $\tilde{\mathbf{x}}^{[]}$, $\tilde{\mathbf{x}}^{[\bullet]_2}, \ldots, \tilde{\mathbf{x}}^{[\bullet]_{t-1}}$ – i.e., $\mathfrak{L}^{[\bullet]_t}\left(\tilde{\mathbf{x}}^{[\bullet]_{t-1}}\right) = \sum_{l_{t-1}=1}^{L_{t-1}} \theta^{[\bullet]_t}$.

Several variants of *FFFB* are possible and a few are discussed after this listing (Birge et al., Gassmann, and Wittrock).

Forwards First (FF): The algorithm moves from nodes in period t to the parent nodes in period $t-1$ only when the solutions for all subproblems in periods $t,\ldots,T$ are optimal (Birge et al. and Gassmann).

Backwards First (BF): The algorithm moves from nodes in period t to the child nodes in period $t+1$ only when no new cuts for period $t-1$ are generated by the nodes in period t (Birge et al., Gassmann, and Wittrock).

Birge et al., Gassmann, and Wittrock all concur that the *FFFB* protocol is generally the most efficient. Birge et al. employ a variant of *FFFB* in which a node in an intermediate period is termed *blocked* during a forward pass if the subproblem at that node is infeasible. The forward pass is then terminated along the path anchored at the blocked node but is continued for all unblocked nodes. Feasibility cuts generated at the blocked nodes are then added to the problems of the parent nodes during the next backward pass. A similar procedure could be used to discontinue a backward pass only along the paths from those nodes that do not generate cuts while continuing to pass back cuts along the remaining paths.

The above descriptions assume that there is a subproblem for each node in each period. The descriptions still apply if certain consecutive periods are grouped into stages. A five-period problem, for example, could have the first period in stage one, periods two and three in stage two, and the remaining two periods in stage three. Such groupings can significantly reduce the number of subproblems but at the expense of increased problem

sizes. These descriptions also apply only to problems that are not block-separable or when the property is not utilized. Strategies for using nested decomposition with block-separable problems are discussed Section 4.4.4 below.

4.4.3 Block-Separable Strategies

Many different decomposition implementation strategies are possible with multi-period problems that have the block-separable property. Strategies based upon arranging the problem structure to resemble a two-period problem are discussed here. Strategies for using nested decomposition with block-separable problems are discussed in the following subsection.

In addition to the different arrangements possible for problem constraints described in Section 4.3, implementation strategies must account for such issues as the number of subproblems and subproblem solution frequency. The number of subproblems determines the number of cuts that are possible – one cut for each subproblem. Note that *subproblem* as used here implies that a single cut is generated for a particular group of constraints as described in Section 3.3.4. Subproblem solution frequency determines the amount of improvement in the upper bound and the frequency of updates to the lower bound.

In order to demonstrate these concepts, define aggregate level constraints as constraints containing only aggregate level, $\mathbf{x}^{(1)[\bullet]_t}$, variables and detailed level constraints as constraints that incorporate detailed level, $\mathbf{x}^{(2)[\bullet]_t}$, variables. Let $\bar{t}$, $1 \leq \bar{t} \leq T$, index the last period containing aggregate level constraints that are brought into the RMP. Then only

aggregate level constraints from the first period are in the RMP if $\bar{t} = 1$ while all aggregate level constraints are in the RMP if $\bar{t} = T$. An example of the latter case is the arrangement of the constraints in problem [4.17]. Detailed level constraints in periods periods $1, \ldots, \bar{t}$ are not in the RMP and form a series of independent subproblems. There can be at least $H^{(\bar{t})} = \sum_{t=1}^{\bar{t}} H_t$ such subproblems, i.e., one subproblem for the detailed level constraints at each node in periods 1 through $\bar{t}$. Fewer subproblems are created if cuts are aggregated and more are possible if the structures of matrices $\mathbf{A}^{(2,2)}$ and $\mathbf{W}^{(2,2)[\bullet]_t}$, $t = 1, \ldots, \bar{t}$, allow for even more separability. Note that the latter requires a modification to the notation since the applicable nodes would then be associated with more than one θ value. Each node in period $\bar{t}+1$ then anchors a subproblem with $T-\bar{t}$ periods if $\bar{t} < T$. Figure 4.3 demonstrates these ideas on a four-period decision tree with binary outcomes.

It is not possible to list all decomposition strategies for a general multi-period block-separable problem. Therefore implementation strategies will be defined by a flexible three-level numeric planning scheme: *major strategy*, *minor strategy*, and *tactics*.

The major strategy defines the structure of the initial RMP and such issues as whether to employ cold (no advance basis) or warm (advanced basis) starts to solve subproblems and the solution frequency for subproblems. For instance, *major strategy = 1* might indicate that $\bar{t} = T - 1$, cold starts are to be used on all subproblems, and all subproblems are to be solved at each decomposition iteration while *major strategy = 2* is the same except warm starts are to be used with subproblems.

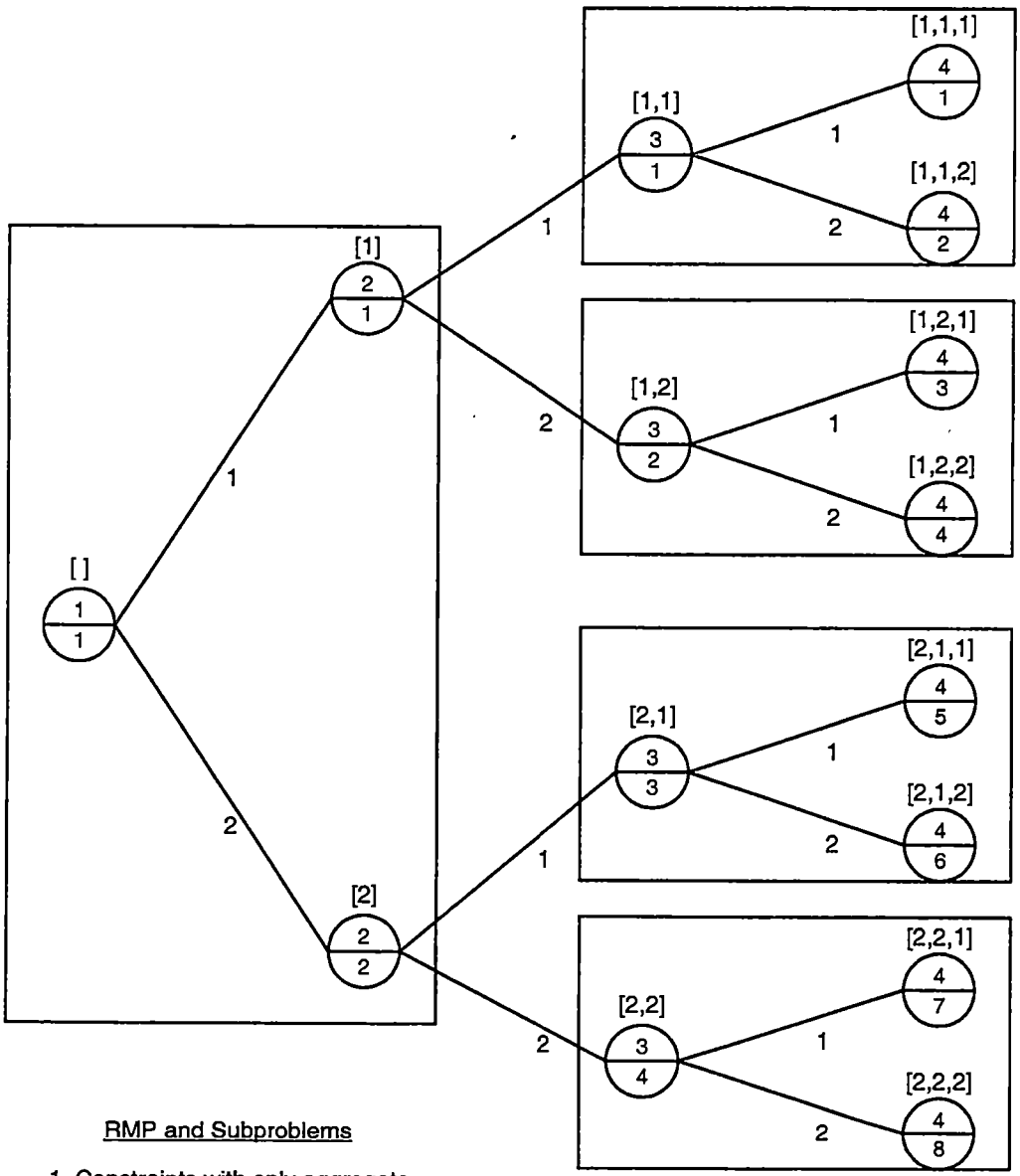


Figure 4.3: Four Period Block-Separable Problem With $\bar{t} = 2$

The number and formulation of subproblems is designated by the minor strategy. In other words, the minor strategy dictates how the constraints left out of the RMP are used to create subproblems and therefore controls the maximum number of cuts that can be added to the RMP at the conclusion of each iteration. For example, *minor strategy = 1* could mean that only one subproblem is to be employed while *minor strategy = 2* indicates that there is a subproblem for each node in period $\bar{t}$ and one subproblem for all remaining constraints.

Finally, the tactics control the number and types of cuts actually added to the RMP at the conclusion of each iteration. For instance, only the cut associated with the maximum violation (e.g., maximum $\theta^{[\bullet]_t} - \mathfrak{L}^{[\bullet]_t}\left(\tilde{\mathbf{x}}^{(1)[\bullet]_t}\right)$ difference) is added when *tactics = 1* while all possible cuts are added if *tactics = 2*.

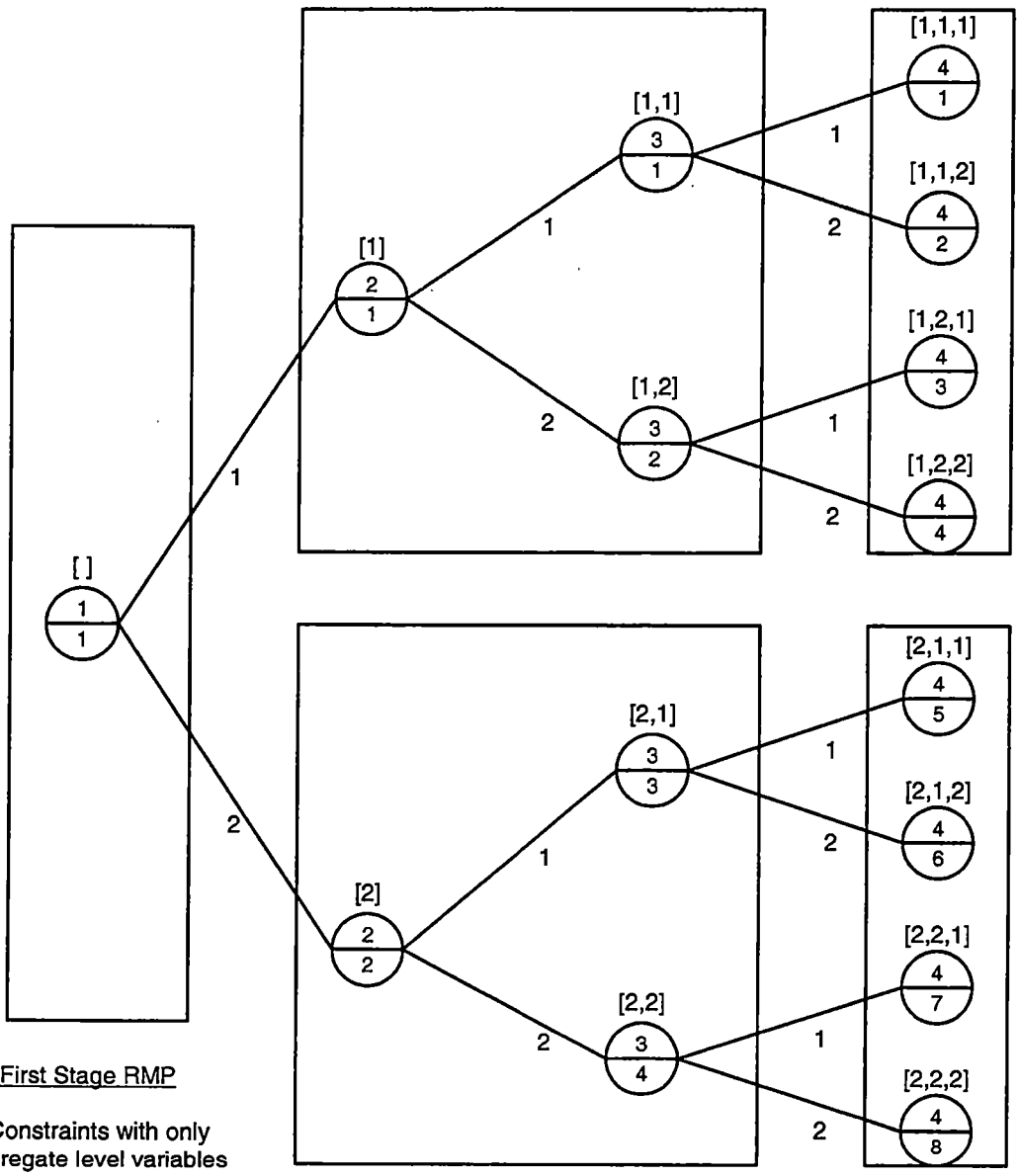
Decomposition method and strategies are then shown as LSD($i.j.k$) or DWD($i.j.k$) where i indicates the major strategy, j the minor strategy, and k the tactics. This scheme is used in Chapter 6 to label the decomposition strategies employed with a multi-period block-separable market investment model. Variations to the scheme are discussed in the following subsection and where necessary in Chapter 6.

4.4.4 Nested Decomposition of Block-Separable Problems

Very large multi-period problems may require nested decomposition even if the problems have the block-separable property. Block-separable problems allow for far more flexibility in the application of nested decomposition than do problems without this property. Block-separability may be applied within each stage of the procedure where it is applicable. For

instance, assume that a four-period problem with binary outcomes is block-separable in every period. Then the periods can be grouped by stages – say, for example, period one in the first stage, periods two and three in the second stage, and period four in the third and last stage. Aggregate level constraints in period (stage) one go into the first stage RMP while the first period detailed level constraints form second stage subproblems. Each second period node would then anchor a three-period second stage RMP-SUB. These problems would act as subproblems to the first stage while in the SUB mode and as second stage relaxed master problems when in the RMP mode. Nested decomposition could be used to solve each of these three-period problems. Aggregate level constraints in periods two and three would go into the second stage RMP while the detailed level constraints in those periods form third stage subproblems. Finally, each fourth period node is treated as a single third stage subproblem. Figure 4.4 diagrams the preceding example.

Nested decomposition could be applied to the above four-period problem in several other ways. The number of ways of using nested decomposition on a block-separable problem grows with the number of periods in the planning horizon. Variations on the implementation strategy scheme described in Section 4.4.3 are required to account for nested decomposition of block-separable multi-period problems. For instance, the grouping of periods into stages and the level of nesting must be defined at one or more of the three strategy levels. These variations are discussed in more detail in the Chapter 6 with the application of nested decomposition to a market investment model.



First Stage RMP

1. Constraints with only aggregate level variables go into first stage RMP
2. Constraints with both aggregate and detailed level variables form second stage subproblems

Subproblems - Nested Decomposition

1. Constraints in periods 2 and 3 with only aggregate level variables form second stage RMP-SUB
2. Constraints in periods 2 and 3 with both aggregate and detailed level variables form third stage subproblems
3. Each fourth period node anchors a third stage subproblem

Figure 4.4: Nested Decomposition of a Block-Separable Four-Period Problem

4.5 Myopic Decomposition

Dantzig-Wolfe and/or L-Shaped decomposition of a multi-period stochastic linear program offer several attractive qualities versus a grand LP solution to the same program. Decomposition methods generally require significantly less memory, especially in early iterations, than do grand LP approaches. Subproblems may be solvable with simple algorithms versus complex simplex algorithms. Decomposition algorithms may be terminated once an ε-optimal solution is available. These qualities generally mean that a decomposition algorithm is much more efficient than a grand LP formulation for large problems.

DWD and/or LSD applied to very large scale problems may, however, require significant amounts of both computer clock time (actual run time) and *central processing unit* (CPU)time. Clock time can be much larger than CPU time on systems that employ some type of *virtual memory* in which a portion of the executable code and problem data may be *paged* to the external memory system. Such a system allows for much larger size problems but frequent requests for paged memory, which does not factor into the CPU time, can greatly increase clock time over CPU time.

One procedure that may provide acceptable results in less time than either DWD or LSD is *myopic* decomposition. The primal form of this procedure takes a short-sighted view of the overall problem by solving decomposed subproblems in each period using information from the previous periods but neglecting any effects the solution has on following periods. The dual form is similar except it solves decomposed subproblems in each period

using information from future periods and neglects effects on previous periods. Either or both the primal and dual forms of myopic decomposition can be used to obtain an approximate solution to the problem. The primal form will provide a lower bound for a problem whose objective is to be maximized while the dual form will provide an upper bound. Myopic decomposition is not new approach – Mossin [150, page 223] (1968) described the concept as a method an investor might employ to maximize expected utility of final wealth in a period while disregarding the future. Khang and Fujiwara [121] recently investigated the use of a myopic view in optimizing ordering policies when there is stochastic supply.

Both the dual and primal forms are used in the *myopic dual-primal cycling algorithm* (**MDPCA**)developed below. Formulations for the decomposed subproblems used by the algorithm are described first. Algorithm **MDPCA** is then described followed by heuristic modifications to the algorithm. Note that the algorithm is developed assuming that the multi-period stochastic linear program is bounded and has complete or relatively complete recourse. Requiring complete recourse is not seen as a serious drawback since, according to Birge and Louveaux [17, Section 3.1(d)], this property is present in most practical problems and is often added to problems without complete recourse. In any case, the algorithm could be easily modified for a general problem by incorporating feasibility and optimality (unbounded) cuts similar to those defined in Chapter 3.

4.5.1 Myopic Subproblems

The subproblems used in myopic decomposition are based upon the grand LP formulations of the primal multi-period problem **PMPGLP** [2.8] on page 35 and the corresponding dual problem **DMPGLP** [2.12] on page 39. A primal phase subproblem at some node in the decision tree assumes a feasible solution vector is available from the parent node and ignores the effect its solution has on all descendent nodes. The resulting subproblem is termed a *nodal primal subproblem* and is represented as a function, $\mathfrak{M}_{\mathrm{P}}^{[\bullet]_t}\left(\mathbf{x}^{[\bullet]_{t-1}}\right)$, of the solution vector from the parent node. The first period nodal primal subproblem is then:

$$\begin{aligned} \mathfrak{M}_{\mathrm{P}}^{[]}(\,) = \max \quad & \mathbf{c}'^{[]}\mathbf{x}^{[]} & & \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x}^{[]} & \leq & \mathbf{b}^{[]}, \\ & \mathbf{x}^{[]} & \geq & \mathbf{0}, \end{aligned} \qquad [4.18]$$

and the nodal primal subproblem at any node in period t, $t = 2, \ldots, T$, is:

$$\begin{aligned} \mathfrak{M}_{\mathrm{P}}^{[\bullet]_t}\left(\mathbf{x}^{[\bullet]_{t-1}}\right) = \hat{p}^{[\bullet]_t} \max \quad & \mathbf{c}'^{[\bullet]_t}\mathbf{x}^{[\bullet]_t} & & \\ \text{s.t.} \quad & \mathbf{W}^{[\bullet]_t}\mathbf{x}^{[\bullet]_t} & \leq & \mathbf{b}^{[\bullet]_t} - \mathbf{B}^{[\bullet]_t}\mathbf{x}^{[\bullet]_{t-1}}, \\ & \mathbf{x}^{[\bullet]_t} & \geq & \mathbf{0}. \end{aligned} \qquad [4.19]$$

A dual phase subproblem at some node in the decision tree assumes a feasible solution vector is available from each child node and ignores the effect its solution has on all ascendant nodes. The resulting subproblem is termed a *nodal dual subproblem* and is represented as a function, $\mathfrak{M}_{\mathrm{D}}^{[\bullet]_t}\left(\left[\boldsymbol{\pi}^{[\bullet]_{t+1}}\right]_{l_t=1}^{L_t}\right)$, of the solution vectors, $\left[\boldsymbol{\pi}^{[\bullet]_{t+1}}\right]_{l_t=1}^{L_t}$, from the child nodes where $\left[\boldsymbol{\pi}^{[\bullet]_{T+1}}\right]_{l_T=1}^{L_T}$ is simply ignored at nodes in the terminal period. The nodal dual subproblem at any node in the terminal period, T, is then:

$$\begin{aligned} \mathfrak{M}_{\mathrm{D}}^{[\bullet]_T}(\,) = \hat{p}^{[\bullet]_T} \min \quad & \mathbf{b}'^{[\bullet]_T}\boldsymbol{\vartheta}^{[\bullet]_T} & & \\ \text{s.t.} \quad & \mathbf{W}'^{[\bullet]_T}\boldsymbol{\vartheta}^{[\bullet]_T} & \geq & \mathbf{c}^{[\bullet]_T}, \\ & \boldsymbol{\vartheta}^{[\bullet]_T} & \geq & \mathbf{0}, \end{aligned} \qquad [4.20]$$

and $\pi^{[\bullet]_T} = \hat{p}^{[\bullet]_T}\tilde{\vartheta}^{[\bullet]_T}$ at each node where $\tilde{\vartheta}^{[\bullet]_T}$ is the optimal solution to problem [4.20] at the corresponding node. The nodal dual subproblem at any node in period, t, $t = T - 1, \ldots, 2$, is:

$$\begin{aligned} \mathfrak{M}_{\mathrm{D}}^{[\bullet]_t}\left(\left[\pi^{[\bullet]_{t+1}}\right]_{l_t=1}^{L_t}\right) = \hat{p}^{[\bullet]_t} \min \quad & \mathbf{b}'^{[\bullet]_t}\vartheta^{[\bullet]_t} \\ \text{s.t.} \quad \mathbf{W}'^{[\bullet]_t}\vartheta^{[\bullet]_t} &\geq \mathbf{c}^{[\bullet]_t} - \tfrac{1}{\hat{p}^{[\bullet]_t}} \sum_{l_t=1}^{L_t} \mathbf{B}'^{[\bullet]_{t+1}}\pi^{[\bullet]_{t+1}}, \\ \vartheta^{[\bullet]_t} &\geq \mathbf{0}, \end{aligned} \qquad [4.21]$$

the nodal dual subproblem at the single first period node is:

$$\begin{aligned} \mathfrak{M}_{\mathrm{D}}^{[]}\left(\left[\pi^{[\bullet]_2}\right]_{l_1=1}^{L_1}\right) = \min \quad & \mathbf{b}'^{[]}\vartheta^{[]} \\ \text{s.t.} \quad \mathbf{A}'\vartheta^{[]} &\geq \mathbf{c}^{[]} - \sum_{l_1=1}^{L_1} \mathbf{B}'^{[\bullet]_2}\pi^{[\bullet]_2}, \\ \vartheta^{[]} &\geq \mathbf{0}, \end{aligned} \qquad [4.22]$$

and $\pi^{[\bullet]_t} = \hat{p}^{[\bullet]_t}\tilde{\vartheta}^{[\bullet]_t}$ at each node where $\tilde{\vartheta}^{[\bullet]_t}$ is the optimal solution to problem [4.21] or problem [4.22] as appropriate at the corresponding node.

4.5.2 Algorithm MDPCA

Algorithm **MDPCA** is an iterative procedure where one pass is made through the nodal primal subproblems and one pass is made through the nodal dual subproblems during each iteration. Each iteration is called a *cycle* and the first pass is termed the *lead half-cycle* while the second pass is referred to as the *tail half-cycle*. Either class of subproblems, dual or primal, may be in the lead half-cycle and the other class in the tail half-cycle. The resulting procedure is said to be the *primals lead* or the *duals lead* version as appropriate.

A primal pass starts with a solution to the first period nodal primal subproblem [4.18]. Solutions are then obtained to the remaining primal subproblems [4.19] in an iterative, breadth-first, node order through the decision tree. Note that the dual multipliers to the period T nodal primal subproblems [4.19] are feasible (after dividing by $\hat{p}^{[\bullet]_T}$) to the period T nodal dual subproblems [4.20]. Therefore, a dual pass following a primal pass can start with the nodal dual subproblems in period $T-1$.

A dual pass starts with solutions to the terminal period nodal dual subproblems [4.20]. Solutions are then obtained to the remaining dual subproblems [4.21] and [4.22] by proceeding backwards through the tree in an iterative node order. Note that the dual multipliers to the first period nodal dual subproblem [4.22] are feasible to the first period nodal primal subproblem [4.18]. Therefore, a primal pass following a dual pass can start with the nodal primal problems in the second period.

Best lower and upper bounds are updated after the completion of each primal half-cycle and each dual half-cycle respectively. There is no guarantee that the bounds will converge. Therefore the algorithm is terminated either when:

1. the relative difference between the bounds is less than or equal to some prespecified positive amount, ε, or
2. the difference between the bounds fails to decrease from one cycle to the next.

The primals lead version of the algorithm is initiated by solving the first period nodal primal subproblem. Cycling iterations then begin with the primal half-cycle starting with

the second period primal subproblems. Information from the period T primal subproblems is used to start the dual half-cycle with the period $T-1$ nodal dual subproblems. Once the first period dual subproblem is solved, information from that problem is used to start the next cycle with the second period nodal primal subproblems. Cycling continues until at least one of the termination criteria is satisfied.

The duals lead version is initiated by solving the period T nodal dual subproblems. Cycling iterations then begin with the dual half-cycle starting with the period $T-1$ dual subproblems. Information from the first period dual subproblem is used to start the primal half-cycle with the second period nodal primal subproblems. Once the terminal period primal subproblems are solved, information from those problems is used to start the next cycle with the period $T-1$ nodal dual subproblems. Cycling continues until at least one of the termination criteria is satisfied.

Myopic decomposition can be an attractive alternative to DWD and LSD even though the latter two methods will give a solution to within an arbitrary tolerance while myopic decomposition will not. All three methods require roughly the same amount of *overhead* memory – maintained storage of such items as best solution to date information and housekeeping variables. Myopic decomposition, on the other hand, generally requires significantly less additional active memory (*random access memory* or RAM) than do DWD and LSD. At the most memory intensive point in the procedure, algorithm **MDPCA** needs access to enough problem data and executable code to solve the largest subproblem. Dantzig-Wolfe and L-Shaped decomposition of large problems will generally require the same in-

formation plus they must maintain the corresponding RMP in active memory. Furthermore, myopic decomposition may arrive at a bounds gap after the first complete cycle that either DWD or LSD would take substantially more time to achieve. Algorithm **MDPCA** is also easily modified to act as an initiating procedure for either of the other two decomposition methods. Empirical results that provide insight into these issues are given in the Chapter 6.

The detailed description of algorithm **MDPCA** is quite lengthy and is placed in Appendix B along with flowcharts of the algorithm and its contained procedures.

4.5.3 Heuristic Modifications

Heuristic modifications to algorithm **MDPCA** are based upon changing the formulations of the tail half-cycle nodal subproblems. Nodal primal subproblems in the duals lead version of the algorithm are the decomposed nodal problems of the dual to problem **DMPGLP** [2.12] modified as follows:

$$\hat{z}^{\mathrm{D}} = \min \mathbf{b}'^{[]}\boldsymbol{\pi}^{[]} + \sum_{l_1}^{L_1}\left[\mathbf{b}'^{[\bullet]_2}\boldsymbol{\pi}^{[\bullet]_2} + \left[\sum_{l_2}^{L_2}\mathbf{b}'^{[\bullet]_3}\boldsymbol{\pi}^{[\bullet]_3} + \left[\ldots + \left[\sum_{l_{T-1}}^{L_{T-1}}\mathbf{b}'^{[\bullet]_T}\boldsymbol{\pi}^{[\bullet]_T}\right]\ldots\right]\right]\right] \qquad [4.23]$$

$$\begin{aligned}
\text{s.t.} \qquad \mathbf{A}'\boldsymbol{\pi}^{[]} &\geq \mathbf{c}^{[]} - \sum_{l_2=1}^{L_2}\mathbf{B}'^{[\bullet]_2}\boldsymbol{\pi}^{[\bullet]_2}, \\
\mathbf{W}'^{[\bullet]_t}\boldsymbol{\pi}^{[\bullet]_t} &\geq \hat{p}^{[\bullet]_t}\mathbf{c}^{[\bullet]_t} - \sum_{l_t=1}^{L_t}\mathbf{B}'^{[\bullet]_{t+1}}\boldsymbol{\pi}^{[\bullet]_{t+1}}, \qquad \begin{matrix} h_t = 1,\ldots,H_t, \\ t = 2,\ldots,T-1, \end{matrix} \\
\mathbf{W}'^{[\bullet]_T}\boldsymbol{\pi}^{[\bullet]_T} &\geq \hat{p}^{[\bullet]_T}\mathbf{c}^{[\bullet]_T}, \qquad h_T = 1,\ldots,H_T, \\
\boldsymbol{\pi}^{[]} &\geq \mathbf{0}, \\
\boldsymbol{\pi}^{[\bullet]_t} &\geq \mathbf{0}, \qquad \begin{matrix} h_t = 1,\ldots,H_t, \\ t = 2,\ldots,T. \end{matrix}
\end{aligned}$$

Nodal dual subproblems in the primals lead version of the algorithm are the decomposed nodal problems of the dual to problem **PMPGLP** [2.8] modified as follows:

$$\begin{aligned}
\hat{z}^{\mathrm{P}} = \max \ \ & \mathbf{c}'^{[]}\mathbf{x}^{[]} + \sum_{l_1=1}^{L_1} p_{l_1}^{[]} \left[\mathbf{c}'^{[\bullet]_2}\mathbf{x}^{[\bullet]_2} + \sum_{l_2=1}^{L_2} p_{l_2}^{[\bullet]_2} \left[\mathbf{c}'^{[\bullet]_3}\mathbf{x}^{[\bullet]_3} + \ldots + \right.\right. \\
& \left.\left. \sum_{l_{T-1}=1}^{L_{T-1}} p_{l_{T-1}}^{[\bullet]_{T-1}} \left[\mathbf{c}'^{[\bullet]_T}\mathbf{x}^{[\bullet]_T}\right] \ldots\right]\right] \\
\text{s.t.} \quad & \mathbf{A}\mathbf{x}^{[]} \leq \mathbf{b}^{[]}, \\
& \mathbf{W}^{[\bullet]_t}\mathbf{x}^{[\bullet]_t} \leq \mathbf{b}^{[\bullet]_t} - \mathbf{B}^{[\bullet]_t}\mathbf{x}^{[\bullet]_{t-1}}, \quad h_t = 1, \ldots, H_t, \quad t = 2, \ldots, T, \\
& \mathbf{x}^{[]} \geq \mathbf{0}, \\
& \mathbf{x}^{[\bullet]_t} \geq \mathbf{0}, \qquad\qquad h_t = 1, \ldots, H_t, \quad t = 2, \ldots, T.
\end{aligned} \tag{4.24}$$

The primal subproblem for the first period is not required in the duals lead version of algorithm **MDPCA**. Heuristically modified nodal primal subproblems for nodes in periods $2, \ldots, T-1$ for the duals lead version are the decomposed nodal subproblems of the dual to problem [4.23]:

$$\begin{aligned}
\widehat{\mathfrak{M}}_{\mathrm{P}}^{[\bullet]_t}\left(\mathbf{x}^{[\bullet]_{t-1}}\right) = \hat{p}^{[\bullet]_t} \max \ \ & \left(\mathbf{c}^{[\bullet]_t} - \tfrac{1}{\hat{p}^{[\bullet]_t}} \sum_{l_t=1}^{L_t} \mathbf{B}'^{[\bullet]_{t+1}}\boldsymbol{\pi}^{[\bullet]_{t+1}}\right)' \mathbf{x}^{[\bullet]_t} \\
\text{s.t.} \quad & \mathbf{W}^{[\bullet]_t}\mathbf{x}^{[\bullet]_t} \leq \mathbf{b}^{[\bullet]_t} - \mathbf{B}^{[\bullet]_t}\mathbf{x}^{[\bullet]_{t-1}}, \\
& \mathbf{x}^{[\bullet]_t} \geq \mathbf{0},
\end{aligned} \tag{4.25}$$

and the terminal period primal subproblems remain unchanged from problems [4.19]. Dual subproblems for the terminal period are not required in the primals lead version of the algorithm. Heuristically modified nodal dual subproblems for nodes in periods $T-1, \ldots, 2$ in the primals lead version are the decomposed nodal subproblems of the dual to problem

[4.24]:

$$\widehat{\mathfrak{M}}_{\mathrm{D}}^{[\bullet]_t}\left(\left[\pi^{[\bullet]_{t+1}}\right]_{l_t=1}^{L_t}\right) =$$

$$\begin{aligned} \hat{p}^{[\bullet]_t} \min \quad & \left(\mathbf{b}^{[\bullet]_t} - \mathbf{B}^{[\bullet]_t}\mathbf{x}^{[\bullet]_{t-1}}\right)' \vartheta^{[\bullet]_t} \\ \text{s.t.} \quad & \mathbf{W}'^{[\bullet]_t}\vartheta^{[\bullet]_t} \geq \mathbf{c}^{[\bullet]_t} - \tfrac{1}{\hat{p}^{[\bullet]_t}} \sum_{l_t=1}^{L_t} \mathbf{B}'^{[\bullet]_{t+1}}\pi^{[\bullet]_{t+1}}, \\ & \vartheta^{[\bullet]_t} \geq \mathbf{0}, \end{aligned} \qquad [4.26]$$

and the first period dual subproblem remains unchanged from problem [4.18].

Objective function values must be corrected after the solutions are obtained at each node in each applicable period as follows:

$$\begin{aligned} \mathfrak{M}_{\mathrm{P}}^{[\bullet]_t}\left(\mathbf{x}^{[\bullet]_{t-1}}\right) &= \widehat{\mathfrak{M}}_{\mathrm{P}}^{[\bullet]_t}\left(\mathbf{x}^{[\bullet]_{t-1}}\right) + \mathbf{x}'^{[\bullet]_t} \sum_{l_t=1}^{L_t} \mathbf{B}'^{[\bullet]_{t+1}}\pi^{[\bullet]_{t+1}} \\ \mathfrak{M}_{\mathrm{D}}^{[\bullet]_t}\left(\left[\pi^{[\bullet]_{t+1}}\right]_{l_t=1}^{L_t}\right) &= \widehat{\mathfrak{M}}_{\mathrm{D}}^{[\bullet]_t}\left(\left[\pi^{[\bullet]_{t+1}}\right]_{l_t=1}^{L_t}\right) + \pi'^{[\bullet]_t}\mathbf{B}^{[\bullet]_t}\mathbf{x}^{[\bullet]_{t-1}} \end{aligned}$$

Nodal subproblems [4.25] and [4.26] are not valid formulations since the problems they are based upon, [4.23] and [4.24] respectively, are invalid as formulated. Each of the latter two problems treats some variables as unknown in the objective function while treating these same variables as known in the constraints. These unknown-known variables then become part of the objective function coefficients in the nodal subproblems. Since each set of resulting subproblems are in the tail half-cycle of the algorithm, the corresponding variables have been assigned values during the lead half-cycle. The heuristic method proved to be more effective than the unmodified algorithm on the market investment model de-

veloped in Chapter 5. No theoretical evidence is currently available to suggest that the heuristically modified algorithm would be more effective on a general class of problems.

4.6 Stochastic Data Storage and Retrieval

Frequent access to stochastic data is a common requirement of each of the decomposition methods examined herein. Efficient data storage and retrieval procedures are therefore an imperative component of any effective decomposition algorithm. A Fortran 95 module, referred to as *Stoc_Forest*, was developed to implement these procedures. This module encapsulates the data structures and functions required to efficiently store and retrieve stochastic data.

Data structures in module *Stoc_Forest* are based upon the structure of the decision tree. Problem data are considered to be assigned to the nodes of the tree. A block of nodal data is either stored or retrieved by traversing the tree with pointers directing traffic from one node to another. Individual nodes are identified by the path vector, $[\bullet]_t = [l_1, \ldots, l_{t-1}]$, to the desired node. Functions are provided to store data by node in either an iterative (breadth-first) or recursive (depth-first) order. Both storage orders require that data for all ancestral nodes be stored prior to storing data for a descendant node. Stored data may be retrieved in any nodal order and may be retrieved as a block (all data for a given node) or by array components.

Module *Stoc_Forest* allows data to be maintained in active memory or to be stored in binary direct access files on a nodal basis. In other words, data for some nodes may be maintained in RAM while data for the other nodes is kept in files. Storing data in files allows for much larger problems at the expense of significantly decreased storage and retrieval efficiencies. The module will also store and retrieve data for a theoretically unlimited number of independent decision trees. The module is very efficient, generally requiring a minuscule amount of time relative to the other processes involved in decomposition procedures.

Chapter 5
Market Investment Model

Decomposition procedures developed in Chapters 3 and 4 are applied to a multi-period market investment model developed by Edirisinghe, [64] and [66], for implementation at a private investment company. Model development, model properties, and solution procedures are described in this chapter. Solution results and analyses are presented in the following chapter.

The market investment model is presented in detail in the first section. Model development incorporates both primal and dual problem formulations. Required array notation and properties of model problem instances are described in the second and third sections respectively. Application of Dantzig-Wolfe, L-Shaped, and myopic decomposition to the model is covered in the fourth section. Computer implementation of the model is the topic of the fifth section. Scenario generation procedures used to simulate uncertain data for the model are detailed in the sixth and final section.

5.1 Model Development

The goal of the market investment model is to optimize the period-to-period holdings in a finite number of securities while accounting for the transaction costs of trades and a specified measure of aversion, or tolerance, to risk. Securities is used herein as a generic term for financial instruments that may include, but not be limited to, such items as stocks,

bonds, and mutual funds. Each type of security is quantified in terms of units defined by the investor. One unit of stock *XYZ*, for example, may equate to 100 shares of that stock. The number of units of an individual security that are held or traded is also referred to as the number of positions held or traded.

The model is formulated as a multiple period stochastic linear program, **MIMPSLP**, to optimize the number of positions of each security to be held in each period of the planning horizon subject to the uncertainties of future events. Optimality of holdings is based upon maximizing the expected net return over the planning horizon subject to penalties assigned during each period for any violations to a specified risk aversion measure.

Notation is first established for a single-period model and is subsequently expanded to account for a model with one or more periods. Let N represent the number of securities and let $n \in \mathsf{N} = \{1, \ldots, N\}$ index a specific security. The performance of a given security, $n \in \mathsf{N}$, is measured by:

1. the expected return, μ_n, of each position held in that security,

2. the positive standard deviation, σ_n, of the uncertain return, and

3. the covariances, σ_{nm}, between security n and securities $m = 1, \ldots, N$, $m \neq n$.

Let $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_n)'$ be the vector of standard deviations and let $\Sigma \in \mathbb{R}^{N \times N}$ represent the covariance matrix which is assumed to be positive definite. Parameters $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_N)'$ and Σ are then assumed to describe a finite joint probability distribution on returns per position. Let L designate the number of possible outcomes in the joint probability space denoted by the outcomes matrix $\mathbf{R} \in \mathbb{R}^{N \times L}$ where $\mathbf{R} = [\mathbf{R}_{\bullet 1}, \ldots, \mathbf{R}_{\bullet l}, \ldots, \mathbf{R}_{\bullet L}]$ and $\mathbf{R}_{\bullet l} =$

$(R_{1l}, \ldots, R_{nl}, \ldots, R_{Nl})'$ is the random vector of individual returns, R_{nl}, for each security $n \in N$ at outcome $l \in \mathsf{L} = \{1, \ldots, L\}$. Note that the transpose of the outcomes matrix, $\mathbf{R}'$, is generally called an observations matrix and $\mathbf{R}'_{n\bullet}$ is the $L \times 1$ observations vector for security $n \in \mathsf{N}$. Vector $\mathbf{p} = (p_1, \ldots, p_l, \ldots, p_L)'$ represents the probabilities of observing each outcome $l = 1, \ldots, L$. Mean returns can then be represented as:

$$\mu_n = \mathcal{E}\left(\mathbf{R}'_{n\bullet}\right) = \mathbf{p}'\mathbf{R}'_{n\bullet} = \sum_{l=1}^{L} p_l R_{nl}, n = 1, \ldots, N,$$

where $\mathcal{E}$ denotes the expectation operator.

Model **MIMPSLP** then consists of two primary components: expected net return and risk aversion. Procedures for incorporating these two components are described in the first two subsections below using the notation established above for a single-period planning horizon. A single-period model formulation follows the descriptions of expected return and risk aversion. The final subsection defines the notation and formulates the model for a general planning horizon with one or more periods.

5.1.1 Expected Return

Expected net return for a given security is determined by subtracting the fixed transaction cost associated with a trade in that security from the total return expected to be observed as a result of the trade. Transaction costs, e.g. commissions, are termed *slippage* and are a function of the trade amount. Notation and procedures for incorporating expected net return in a single-period planning horizon are developed below.

Expected Total Return

Let $\mathbf{y} = (y_1, \ldots, y_N)$ denote the *portfolio variables*, the vector of the number of positions held in each security at the conclusion of all trades at the beginning of the planning horizon. The expected total return, $\mathcal{E}(TR_n)$, for a given security is then:

$$\mathcal{E}(TR_n) = \mathcal{E}(y_n \mathbf{R}'_{n\bullet}) = y_n \mathcal{E}(\mathbf{R}'_{n\bullet}) = \mu_n y_n, n \in \mathsf{N}, \quad [5.1]$$

and the expected cumulative total return, $\mathcal{E}(TR)$, for all securities is:

$$\mathcal{E}(TR) = \mathcal{E}\left(\sum_{n=1}^{N} y_n \mathbf{R}'_{n\bullet}\right) = \sum_{n=1}^{N} \mathcal{E}(y_n \mathbf{R}'_{n\bullet}) = \sum_{n=1}^{N} y_n \mathcal{E}(\mathbf{R}'_{n\bullet}) = \sum_{n=1}^{N} \mu_n y_n = \boldsymbol{\mu}' \mathbf{y}. \quad [5.2]$$

Slippage

Let $\mathbf{y}^{(0)} = \left(y_1^{(0)}, \ldots, y_N^{(0)}\right)$ denote the vector of the number of positions held in each security prior to conducting any trades at the beginning of the planning horizon. Transaction costs associated with a trade in security $n \in \mathsf{N}$ are a function of the amount, $X_n^{(1)}$, of the transaction, i.e., the absolute difference in the number of held positions:

$$X_n^{(1)} = \left| y_n - y_n^{(0)} \right|.$$

Two functions, called slippage functions, are used to determine the transaction costs associated with trades in each security: a slippage per position function and a total slippage function. The slippage per position function, $\overline{\mathcal{S}}_n\left(X_n^{(1)}\right)$, for each security $n \in \mathsf{N}$ is a piece-wise linear concave function with two segments called the first and second slippage regions. Each function $\overline{\mathcal{S}}_n\left(X_n^{(1)}\right)$ is described by four parameters:

1. κ_n: the minimum cost per position,

2. X_n^{cut}: the transaction amount that is the boundary between slippage regions,

3. a_{1n}: the slope of the slippage per position linear function in the first slippage region, and

4. a_{2n}: the slope adjustment factor that is added to a_{1n} to obtain the slope in the second slippage region.

These parameters are restricted by the following conditions for all securities $n = 1, \ldots, N$:

$$0 \leq \kappa_n < \infty,$$

$$0 \leq X_n^{\text{cut}} < \infty,$$

$$0 \leq a_{1n} < \infty,$$

$$\begin{cases} 0 \leq a_{2n} < \infty, & \text{if } a_{1n} = 0, \\ -a_{1n} < a_{2n} < \infty, & \text{if } a_{1n} > 0. \end{cases}$$

The last condition on the slope adjustment factor, a_{2n}, precludes the unreasonable possibility that decreasing transaction costs are associated with increasing transaction amounts. Slippage per position function, $\overline{\mathfrak{S}}_n\left(X_n^{(1)}\right)$, for security $n \in \mathsf{N}$ is then:

$$\overline{\mathfrak{S}}_n\left(X_n^{(1)}\right) = \begin{cases} a_{1n}X_n^{(1)} + \kappa_n, & 0 \leq X_n^{(1)} < X_n^{\text{cut}}, \\ \left(a_{1n} + a_{2n}\right)X_n^{(1)} + \kappa_n - a_{2n}X_n^{\text{cut}}, & X_n^{\text{cut}} \leq X_n^{(1)} < \infty. \end{cases}$$

The total slippage function, $\widetilde{\mathfrak{S}}_n\left(x_n\right)$, for security $n \in \mathsf{N}$ is obtained by multiplying the applicable slippage per position function by the transaction amount, $\widetilde{\mathfrak{S}}_n\left(X_n^{(1)}\right) =$

$X_n^{(1)}\overline{\mathfrak{S}}_n\left(X_n^{(1)}\right)$:

$$\widetilde{\mathfrak{S}}_n\left(X_n^{(1)}\right) = \begin{cases} X_n^{(1)}\left(a_{1n}X_n^{(1)} + \kappa_n\right), & 0 \le X_n^{(1)} < X_n^{\text{cut}}, \\ X_n^{(1)}\left[\left(a_{1n} + a_{2n}\right)X_n^{(1)} + \kappa_n - a_{2n}X_n^{\text{cut}}\right], & X_n^{\text{cut}} \le X_n^{(1)} < \infty. \end{cases} \quad [5.3]$$

Graphs for both functions for a typical security $n \in \mathsf{N}$ are given in Figure 5.1.

Total slippage function $\widetilde{\mathfrak{S}}_n\left(X_n^{(1)}\right)$ is a piece-wise quadratic function and cannot be incorporated as is into a linear program. Each total slippage function [5.3] is therefore approximated with a piece-wise linear (PWL) function of the transaction amount. The PWL function for each security $n = 1, \ldots, N$ consists of $K = K_1 + K_2$ linear functions where K_j is the designated number of PWL segments desired in slippage region j for $j = 1, 2$. Each PWL segment, $k \in \mathsf{K} = \{1, \ldots, K\}$, for security $n \in \mathsf{N}$ is bounded by lower and upper break points $\Psi_{(k-1)n}$ and Ψ_{kn} respectively where $\Psi_{0n} = 0$ and $\Psi_{Kn} = \infty$. The last break point for the first slippage region is assigned to the region boundary, $\Psi_{K_1 n} = X_n^{\text{cut}}$, when $X_n^{\text{cut}} > 0$. Break points Ψ_{kn}, $k = 1, \ldots, K_1 - 1$, in the first slippage region are assigned values based upon either one of two criteria:

1. minimizing the average absolute error between PWL and actual slippage over the first slippage region, or
2. limiting the maximum relative error between PWL and actual slippage in any segment to a specified value $\varsigma_1 > 0$.

Break points Ψ_{kn}, $k = K_1 + 1, \ldots, K$, in the second slippage region are assigned values to limit the maximum relative error in any segment to a specified value $\varsigma_2 > 0$. The linear

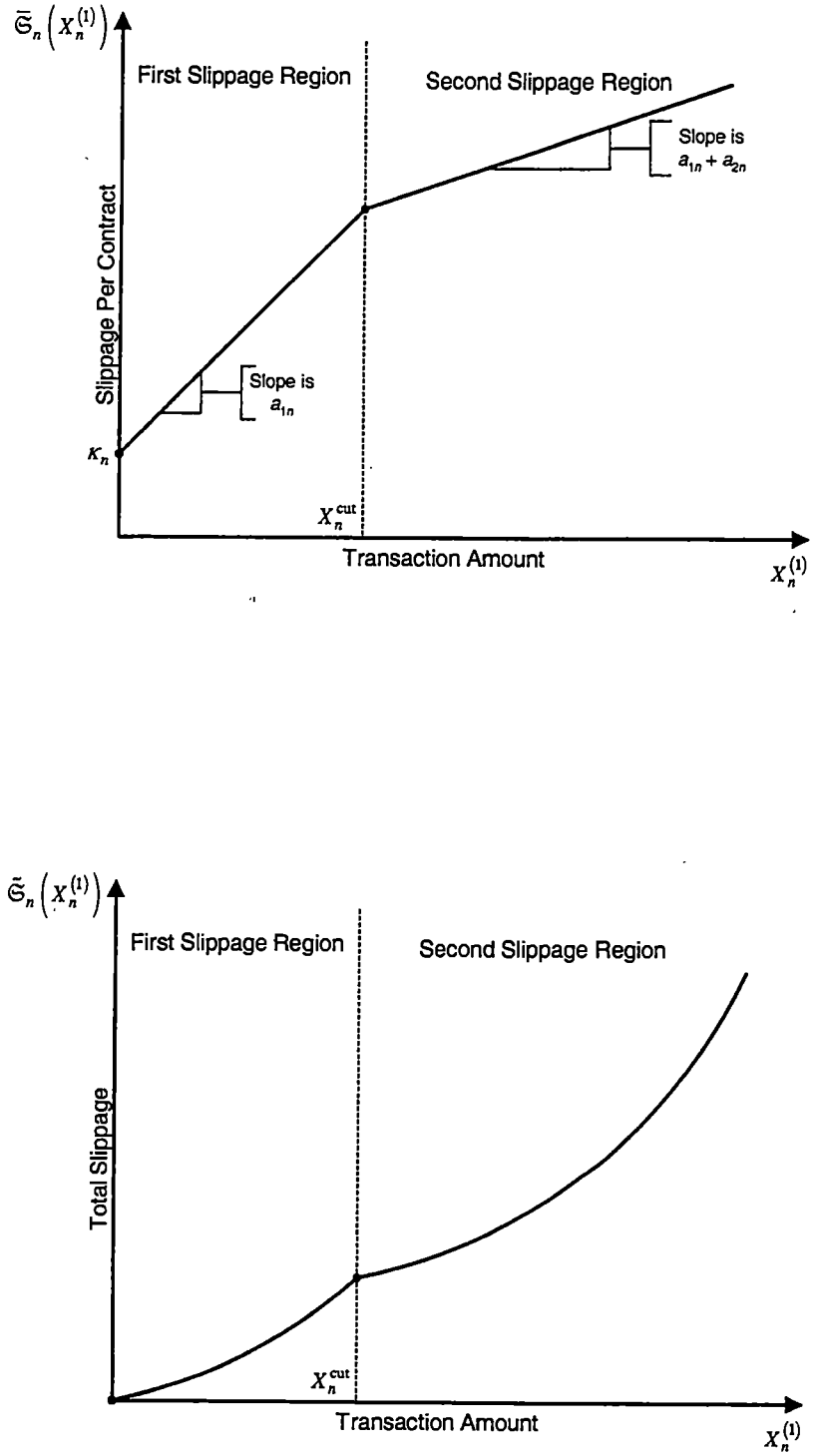


Figure 5.1: Slippage Per Position and Total Slippage for Security $n \in \mathsf{N}$

function in segment $k \in \mathsf{K}$ has slope Φ_{kn} where

$$\Phi_{kn} = \frac{\widetilde{\mathfrak{S}}_n(\Psi_{kn}) - \widetilde{\mathfrak{S}}_n(\Psi_{(k-1)n})}{\Psi_{kn} - \Psi_{(k-1)n}}.$$

All terminal break points Ψ_{Kn} for each security $n = 1, \ldots, N$ are set to positive infinity after the slopes Φ_{Kn} have been determined. Break points and slopes for all securities $n = 1, \ldots, N$ are determined such that:

$$0 = \Psi_{0n} < \Psi_{1n} < \cdots < \Psi_{kn} < \cdots < \Psi_{Kn} = \infty, \qquad [5.4a]$$

$$\Phi_{1n} < \cdots < \Phi_{kn} < \cdots < \Phi_{Kn}. \qquad [5.4b]$$

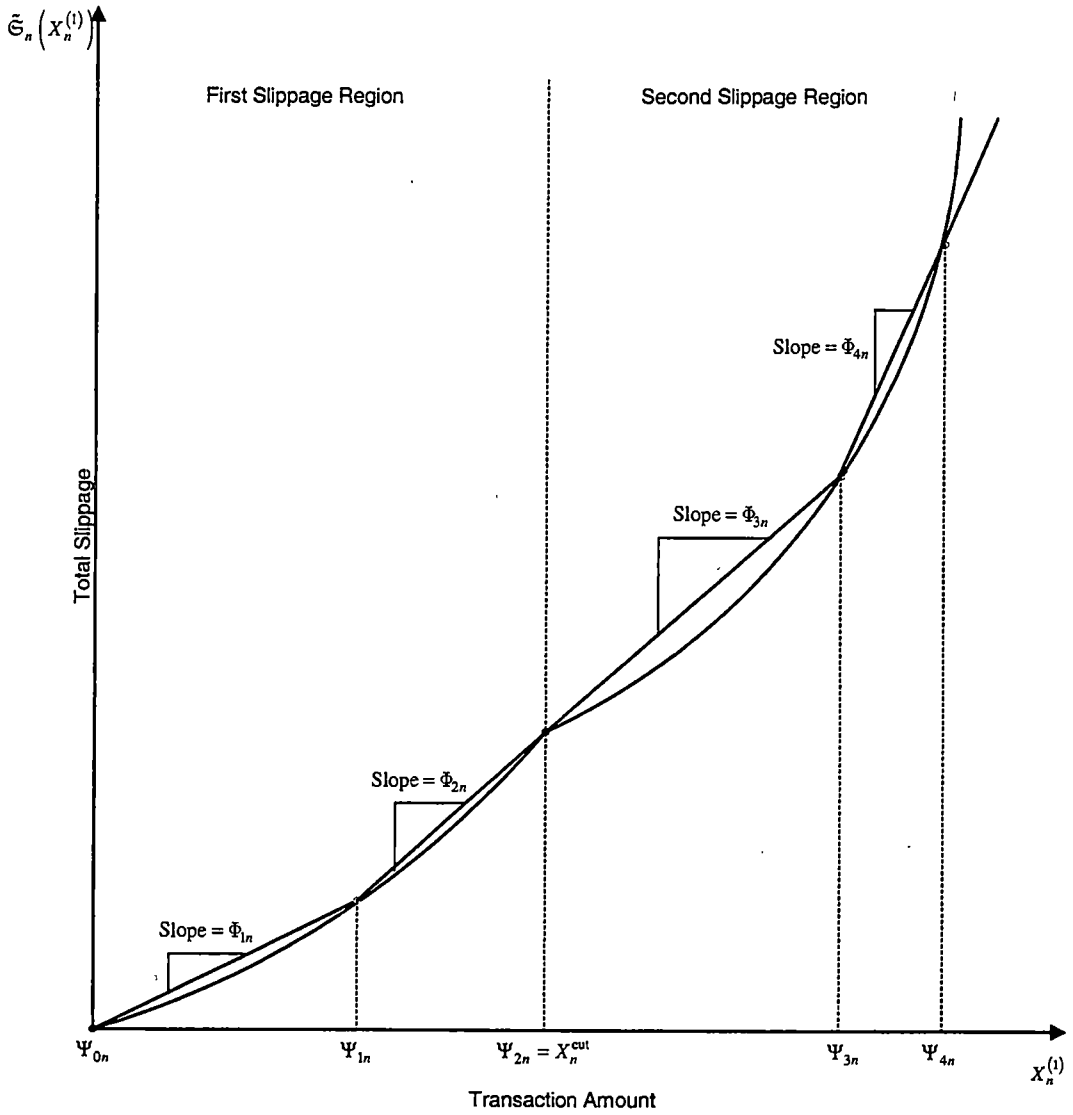
Detailed procedures for assigning values to the PWL break points and slopes are given in Appendix C. A sample graph of the actual and piece-wise linear slippage functions with two PWL segments in each slippage region is illustrated at Figure 5.2.

Expected Net Return

Expected net return for security $n \in \mathsf{N}$ is determined by subtracting the actual slippage defined by equation [5.3] from the expected total return defined by equation [5.1]:

$$\mathcal{E}(NR_n^{\text{act}}) = \mathcal{E}\left[TR_n - \widetilde{\mathfrak{S}}_n\left(X_n^{(1)}\right)\right] = \mathcal{E}(TR_n) - \widetilde{\mathfrak{S}}_n\left(X_n^{(1)}\right) = \mu_n y_n - \widetilde{\mathfrak{S}}_n\left(X_n^{(1)}\right).$$

Model **MIMPSLP** cannot incorporate the above expression, however, since $\widetilde{\mathfrak{S}}_n\left(X_n^{(1)}\right)$ is a piece-wise quadratic function of $X_n^{(1)}$. Instead, the PWL slippage functions described above are used to determine an approximate expected net return for each security. Define



NOTES:

1. Two PWL sections in each slippage region

2. Last PWL break point (Ψ_{4n} in figure) changed to positive infinity after last PWL slope (Φ_{4n}) is determined with:

$$\Phi_{kn} = \frac{\tilde{\Theta}_n\left(\Psi_{kn}\right) - \tilde{\Theta}_n\left(\Psi_{(k-1)n}\right)}{\Psi_{kn} - \Psi_{(k-1)n}},\ k = 1,\ldots,4$$

Figure 5.2: Sample Piece-Wise Linear Slippage Approximation for Security $n \in \mathsf{N}$

the *slippage variables* $X_{kn}^{(1)}$ for each security $n = 1, \ldots, N$ such that:

$$X_n^{(1)} = \sum_{k=1}^{K} X_{kn}^{(1)}, \qquad [5.5a]$$

$$0 \leq X_{kn}^{(1)} \leq \Psi_{kn} - \Psi_{(k-1)n},\ k = 1, \ldots, K. \qquad [5.5b]$$

Then a pair of constraints in addition to those defined by [5.5b] are required for each security to determine $X_{kn}^{(1)}$, $k = 1, \ldots, K$, such that equation [5.5a] is satisfied. The additional constraints for security $n \in \mathsf{N}$ are:

$$y_n - \sum_{k=1}^{K} X_{kn}^{(1)} \leq y_n^{(0)}, \qquad [5.6a]$$

$$y_n + \sum_{k=1}^{K} X_{kn}^{(1)} \geq y_n^{(0)}. \qquad [5.6b]$$

Constraints [5.6a-b] are termed the *slippage constraints* for security $n \in \mathsf{N}$. Constraint [5.6a] is referred to as the slippage *buy* constraint for security n since $y_n > y_n^{(0)}$ if $X_n^{(1)} = \left(y_n - y_n^{(0)}\right)$ while constraint [5.6b] is called the slippage *sell* constraint since $y_n < y_n^{(0)}$ if $X_n^{(1)} = -\left(y_n - y_n^{(0)}\right)$.

Total slippage for security $n \in \mathsf{N}$ is then approximated by:

$$\widetilde{\mathfrak{S}}_n\left(X_n^{(1)}\right) \approx \sum_{k=1}^{K} \Phi_{kn} X_{kn}^{(1)},$$

and approximate expected net return for security n is:

$$\mathcal{E}\left(NR_n^{\text{apx}}\right) = \mu_n y_n - \sum_{k=1}^{K} \Phi_{kn} X_{kn}^{(1)} \qquad [5.7]$$

subject to constraints [5.5b] through [5.6b]. Constraints [5.6a-b] guarantee that

$$\sum_{k=1}^{K} X_{kn}^{(1)} = \left|y_n - y_n^{(0)}\right| = X_n^{(1)}$$

as required by equation [5.5a]. The cumulative expected net return for all securities is then:

$$\mathcal{E}\left(NR^{\text{apx}}\right) = \sum_{n=1}^{N}\left(\mu_n y_n - \sum_{k=1}^{K}\Phi_{kn}X_{kn}^{(1)}\right) = \boldsymbol{\mu}'\mathbf{y} - \sum_{n=1}^{N}\boldsymbol{\Phi}'_{\bullet n}\mathbf{X}_{\bullet n}^{(1)}. \qquad [5.8]$$

5.1.2 Risk Aversion

The model currently incorporates only information on the expected performance of each security without considering the uncertainty of security activity. A portfolio, $\mathbf{y}$, maximizing expected net return would be determined if model construction terminated at this point. This portfolio could, however, result in an actual net return significantly different from that predicted by the current model. Markowitz [140, page 77], in fact, rejected the hypothesis that a model should maximize expected return in favor of an approach that couples expected return with the variance of return. Based upon this mean-variance analysis concept and his work on utility of wealth [141], Markowitz [142] proposed methods that measure the *efficiency* of a portfolio by the ratio of the expected return to the variance of the return. These methods equate risk with the standard deviation (square root of the variance) of the return and reduce risk through variance reduction procedures. Variance reduction methods have at least two major drawbacks: they are modelled with nonlinear programs and are therefore computationally expensive and they punish portfolios with significantly higher returns than expected as well as those with lower returns. Advances in technology as well as linear approximation techniques such as the *mean absolute deviation risk* models of Konno and Yamazaki [128] and Konno and Kobayashi [127] reduce the importance of the

nonlinear programming requirement of variance reduction methods. Punishing portfolios with significantly higher than expected returns as well as those with significantly lower than expected returns is, therefore, considered the more critical of the two shortcomings. One procedure for circumventing the high returns punishment problem is the *semi-variance* concept of Markowitz [142, Chapter 9] – risk is equated to the variance of only those returns that fall below the mean value. Fishburn [78] expands on the semi-variance concept by introducing a range of models in which risk is associated with returns that fall below some target value.

Model **MIMPSLP** incorporates the concept of risk associated with below-target returns by including terms in the objective function that penalize only returns with *downside deviation*, i.e., those returns that fall below the expected return. Models equating risk with downside deviation have been developed for and implemented by large corporations. Eppen, Kipp, and Schrage [73] describe a capacity planning model developed for General Motors that penalizes scenarios that induce profits to fall below a specified target level. Worzel, Vassiadou-Zeniou, and Zenios [210] discuss a fixed-income assets management model implemented by Metropolitan Life Insurance Company that penalizes downside deviations from a fixed-income index. The **MIMPSLP** model also hedges against risk by limiting the number of positions that can be held during any period of the planning horizon. These two risk aversion measures are described below.

Downside Deviation

The downside deviation $\mathfrak{Y}_l(\mathbf{y})$ of portfolio $\mathbf{y}$ relative to outcome $l \in \mathsf{L}$ is defined as:

$$\mathfrak{Y}_l(\mathbf{y}) = \max\left(\left[\left(\boldsymbol{\mu}'\mathbf{y} - \sum_{n=1}^{N} \boldsymbol{\Phi}'_{\bullet n}\mathbf{X}^{(1)}_{\bullet n}\right) - \left(\mathbf{R}'_{\bullet l}\mathbf{y} - \sum_{n=1}^{N} \boldsymbol{\Phi}'_{\bullet n}\mathbf{X}^{(1)}_{\bullet n}\right)\right], 0\right)$$

$$\mathfrak{Y}_l(\mathbf{y}) = \max\left[(\boldsymbol{\mu} - \mathbf{R}_{\bullet l})'\,\mathbf{y}, 0\right] \qquad [5.9]$$

Therefore, the downside deviation of a given portfolio relative to a specific outcome is positive only if the net return (or, equivalently, the total return) of the portfolio determined with the specified outcomes vector is less than, or *downside* of, the expected net (total) return. The model incorporates downside deviation to control risk by penalizing the objective function in either one of two modes:

***linear downside deviation*:** the penalty increases with the expected value of the downside deviation, or

***quadratic downside deviation*:** the penalty increases with the expected value of the squared downside deviation - analogous to Markowitz's semi-variance concept.

The quadratic downside deviation mode requires piece-wise linear approximations to the function $\mathfrak{F}[\mathfrak{Y}_l(\mathbf{y})] = \mathfrak{Y}_l^2(\mathbf{y})$. Procedures discussed above for approximating slippage are easily adapted to determine PWL approximation parameters for $\mathfrak{F}[\mathfrak{Y}_l(\mathbf{y})]$ since this function can be considered as a special case of the slippage function for some security with $a_{1n} = 1$ and $a_{2n} = \kappa_n = 0$. The minimum average absolute error procedure can be applied if desired by creating two artificial downside deviation regions with a simulated value for a region boundary analogous to X_n^{cut}. Methods described in Appendix C are then used to

determine break points φ_q, $q = 0, \ldots, Q$, and slopes γ_q, $q = 1, \ldots, Q$, for a specified number Q of PWL segments such that:

$$0 = \varphi_0 < \varphi_1 < \cdots < \varphi_q < \cdots < \varphi_Q = \infty, \qquad [5.10a]$$

$$\gamma_1 < \cdots < \gamma_q < \cdots < \gamma_Q. \qquad [5.10b]$$

Note that the linear downside deviations mode is represented exactly by one segment with the two break points, $\varphi_0 = 0$ and $\varphi_1 = \infty$, and slope $\gamma_1 = 1$.

Define the *deviation variables* $X_{ql}^{(2)}$ for each outcome $l = 1, \ldots, L$ such that:

$$\mathfrak{D}_l(\mathbf{y}) = \sum_{q=1}^{Q} X_{ql}^{(2)}, \qquad [5.11a]$$

$$0 \leq X_{ql}^{(2)} \leq \varphi_q - \varphi_{q-1}, q = 1, \ldots, Q. \qquad [5.11b]$$

Then another constraint in addition to those defined by [5.11b] is required for each outcome to determine $X_{ql}^{(2)}$, $q = 1, \ldots, Q$, such that equation [5.11a] is satisfied. The additional constraint for outcome $l \in \mathsf{L}$, based upon equation [5.9] and termed a *deviation constraint*, is:

$$\sum_{n=1}^{N} (\mu_n - R_{nl})\, y_n - \sum_{q}^{Q} X_{ql}^{(2)} \leq 0. \qquad [5.12]$$

The penalty term in the objective function is then:

$$-\lambda \sum_{l=1}^{L} p_l \left(\sum_{q=1}^{Q} \gamma_q X_{ql}^{(2)} \right), \qquad [5.13]$$

subject to constraints [5.11b] and [5.12]. Risk aversion factor $\lambda > 0$ in penalty term [5.13] is used to model different levels of aversion to risk which increases with λ. Constraints [5.12] guarantee that equation [5.11a] is satisfied.

Limiting Holdings

An additional hedge against uncertainty in performance is provided with the following *risk budget constraint*:

$$\boldsymbol{\sigma}'\mathbf{y} = \sum_{n=1}^{N} \sigma_n y_n \leq \beta_1, \qquad [5.14]$$

where β_1, $0 < \beta_1 < \infty$, is some specified amount of capital. The budget constraint is intended to limit liability if the return for each security drops by an amount equal to the standard deviation of the return. Portfolio $\mathbf{y}$ is also bounded by $\mathbf{Y}_{\bullet 1}^{\min} = (Y_{11}^{\min}, \ldots, Y_{n1}^{\min}, \ldots, Y_{N1}^{\min})'$ and $\mathbf{Y}_{\bullet 1}^{\max} = (Y_{11}^{\max}, \ldots, Y_{n1}^{\max}, \ldots, Y_{N1}^{\max})'$ such that:

$$-\infty < \mathbf{Y}_{\bullet 1}^{\min} \leq \mathbf{y} \leq \mathbf{Y}_{\bullet 1}^{\max}. \qquad [5.15]$$

5.1.3 Single-Period Planning Horizon

Model **MIMPSLP** with a single-period planning horizon then consists of the expected net return [5.8] and penalty term [5.13] in the objective function, slippage constraints [5.6a-b], deviation constraints [5.12], budget constraint [5.14], and bounding constraints, [5.5b], [5.11b], and [5.15]. The primal formulation for the single-period problem is then:

$$z = \max \sum_{n=1}^{N} \mu_n y_n - \sum_{n=1}^{N} \sum_{k=1}^{K} \Phi_{kn} X_{kn}^{(1)} - \lambda \sum_{l=1}^{L} p_l \sum_{q=1}^{Q} \gamma_q X_{ql}^{(2)} \qquad [5.16]$$

$$
\begin{aligned}
\text{s.t.} \quad & \sum_{n=1}^{N} \sigma_n y_n && && \leq \beta_1, \\
& y_n - \sum_{k=1}^{K} X_{kn}^{(1)} && && \leq y_n^{(0)}, \quad n = 1, \ldots, N, \\
& y_n + \sum_{k=1}^{K} X_{kn}^{(1)} && && \geq y_n^{(0)}, \quad n = 1, \ldots, N, \\
& \sum_{n=1}^{N} (\mu_n - R_{nl})\, y_n && - \sum_{q=1}^{Q} X_{ql}^{(2)} && \leq 0, \quad l = 1, \ldots, L,
\end{aligned}
$$

$$
\begin{aligned}
& Y_{n1}^{\min} \leq y_n \leq Y_{n1}^{\max}, && n = 1, \ldots, N \\
& 0 \leq X_{kn}^{(1)} \leq \Psi_{kn} - \Psi_{(k-1)n}, && k = 1, \ldots, K, n = 1, \ldots, N, \\
& 0 \leq X_{ql}^{(2)} \leq \varphi_q - \varphi_{q-1}, && q = 1, \ldots, Q, l = 1, \ldots, L.
\end{aligned}
$$

The single-period problem [5.16] serves primarily to provide the foundation for the formulation of the problem for a general planning horizon.

5.1.4 General Planning Horizon

Formulations are developed below for a general planning horizon, i.e., a horizon with one or more periods. Notation is extended to accommodate a problem with multiple periods followed by formulations for the primal and dual problems.

Let T denote the number of periods where $1 \leq T < \infty$. Transaction costs are assumed to remain constant across all periods. Constant transaction costs, or slippage, is a reasonable assumption given the relative short period lengths, from a few days to a few weeks, that will be examined. Multiple period problems are constructed and their components labelled using the decision tree structure and notation described in Section 2.2. For example, $\mathbf{y}^{[\bullet]_t}$ is the portfolio vector for a period t, $1 \leq t \leq T$, node $[\bullet]_t =$

$[l_1, \ldots, l_{t-1}] = (t, h_t)$ where $[\bullet]_1 = [\,] = (1,1)$ is the single first period node. Equation [2.7] on page 28 is used to relate a period-index node label, (t, h_t), with the corresponding path vector label, $[\bullet]_t = [l_1, \ldots, l_{t-1}]$, throughout this chapter. Let β_t, λ_t, $\mathbf{Y}^{\min}_{\bullet t} \in \mathbb{R}^{N\times 1}$, and $\mathbf{Y}^{\max}_{\bullet t} \in \mathbb{R}^{N\times 1}$ such that

$$\left.\begin{array}{ccccc} 0 & < & \beta_t & < & \infty \\ 0 & < & \lambda_t & & \\ -\infty & < & \mathbf{Y}^{\min}_{\bullet t} & \leq & \mathbf{Y}^{\max}_{\bullet t} \end{array}\right\} t = 1, \ldots, T$$

be the right-hand-sides of the budget constraints, the risk aversion factors, and the lower and upper portfolio bounds respectively for nodes in a specified period. Define ϱ_t, such that $0 < \varrho_t \leq 1$, as the capital discount factor for period t, $1 \leq t \leq T$, where $\varrho_1 = 1$. Let $\hat{\varrho}_t$ be the compound discount factor for period t, $t = 1, \ldots, T$:

$$\hat{\varrho}_t = \prod_{j=1}^{t} \varrho_j, t = 1, \ldots, T.$$

Table 5.1 summarizes the notation used in the formulation of model **MIMPSLP** problems. The formulation of the single-period problem [5.16] can now be extended to accommodate a general planning horizon.

Primal Formulation

Note that slippage for a node in period $t = 2, \ldots, T$ is a function of $\left|\mathbf{y}^{[\bullet]_t} - \mathbf{y}^{[\bullet]_{t-1}}\right|$ where $[\bullet]_t \in \mathsf{H}^{\mathsf{C}}_t\left([\bullet]_{t-1}\right)$, i.e., node $[\bullet]_t$ is a child of node $[\bullet]_{t-1}$. Then, recalling from Section 2.2.2 that $\hat{p}^{[\bullet]_t}$ represents the probability that the process enters the period t node

Table 5.1: Model **MIMPSLP** Notation (continued on next page)

Symbol	Description
a_{1n}	Slope of slippage per position linear function in first region: $0 \leq a_{1n} < \infty$
a_{2n}	Slippage slope adjustment factor for the second region: $-a_{1n} < a_{2n} < \infty$
H_t	Number of nodes in period t: $H_t = \prod_{j=1}^{t-1} L_j$
$H^{(t)}$	Cumulative number of nodes through period t: $H^{(t)} = \sum_{j=1}^{t} H_j$
K_j	Number of slippage PWL segments in slippage region $j = 1, 2$: $0 \leq K_j < \infty$
K	Total number of slippage PWL segments: $K = K_1 + K_2$
k	Index for slippage PWL segment: $k \in \{1, \ldots, K\}$
L_t	Number of outcomes at each node in period t: $2 \leq L_t < \infty$
l_t	Index for a specific outcome: $l_t \in \{1, \ldots, L_t\}$
N	Number of securities in the market investment model analysis: $1 \leq N < \infty$
N	Set of security indices: $\mathsf{N} = \{1, \ldots, N\}$
n	Index for a specific security: $n \in \mathsf{N}$
$\mathbf{p}^{[\bullet]_t}$	Vector of outcomes' probabilities at node $[\bullet]_t$: $\mathbf{p}^{[\bullet]_t} \in \mathbb{R}^{L_t}$
$p_{l_t}^{[\bullet]_t}$	Probability of outcome l_t at node $[\bullet]_t$: $0 < p_{l_t}^{[\bullet]_t} \leq 1$
$\hat{p}^{[\bullet]_t}$	Compound probability of entering node $[\bullet]_t = \left[\tilde{l}_1, \ldots, \tilde{l}_{t-1}\right]$: $\hat{p}^{[\bullet]_t} = \prod_{j=1}^{t-1} p_{\tilde{l}_j}^{[\bullet]_j}$
Q	Number of downside deviation PWL segments: $1 \leq Q < \infty$
q	Index for downside deviation PWL segment: $q \in \{1, \ldots, Q\}$
$\mathbf{R}^{[\bullet]_t}$	Outcomes matrix of returns at node $[\bullet]_t$: $\mathbf{R}^{[\bullet]_t} \in \mathbb{R}^{N \times L_t}$
$R_{nl_t}^{[\bullet]_t}$	Return for security n given outcome l_t at node $[\bullet]_t$: $-\infty < R_{nl_t}^{[\bullet]_t} < \infty$
T	Number of periods in the planning horizon: $1 \leq T < \infty$
t	Index for a specific period: $t \in \{1, \ldots, T\}$
X_n^{cut}	Boundary between slippage regions for security n: $0 \leq X_n^{\text{cut}} < \infty$
$X_n^{(1)[\bullet]_t}$	Transaction amount for security n at node $[\bullet]_t$: $X_n^{(1)[\bullet]_t} = \left\lvert y_n^{[\bullet]_t} - y_n^{[\bullet]_{t-1}} \right\rvert$
$X_{kn}^{(1)[\bullet]_t}$	Portion of transaction amount for security n at node $[\bullet]_t$ in PWL segment k
$X_{ql_t}^{(2)[\bullet]_t}$	Portion of downside deviation for outcome l_t at node $[\bullet]_t$ in PWL segment q
$\mathbf{Y}^{\max}$	Matrix of maximum number of positions: $\mathbf{Y}^{\max} \in \mathbb{R}^{N \times T}$
$Y_{nt}^{\max}$	Maximum number of positions for security n in period t: $-\infty < Y_{nt}^{\max}$
$\mathbf{Y}^{\min}$	Matrix of minimum number of positions: $\mathbf{Y}^{\min} \in \mathbb{R}^{N \times T}$
$Y_{nt}^{\min}$	Minimum number of positions for security n in period t: $-\infty < Y_{nt}^{\min} \leq Y_{nt}^{\max}$
$\mathbf{y}^{[\bullet]_t}$	Vector of portfolio decision variables at node $[\bullet]_t$: $\mathbf{y}^{[\bullet]_t} \in \mathbb{R}^N$
$y_n^{[\bullet]_t}$	Portfolio decision variable for security n at node $[\bullet]_t$: $Y_{nt}^{\min} \leq y_n^{[\bullet]_t} \leq Y_{nt}^{\max}$
$\mathbf{y}^{(0)}$	Vector of number of positions held at beginning of planning horizon: $\mathbf{y}^{(0)} \in \mathbb{R}^N$
$y_n^{(0)}$	Number of positions held at beginning of planning horizon in security n: $-\infty < y_n^{(0)}$

Table 5.1: (continued) Model **MIMPSLP** Notation

Symbol	Description
β_t	Capital budgeted for risk in period t: $0 < \beta_t < \infty$
γ_q	Downside deviation PWL slope in segment q: $\gamma_1 < \cdots < \gamma_q < \cdots < \gamma_Q$
κ_n	Minimum cost per position: $0 \leq \kappa_n < \infty$
λ_t	Risk aversion factor for period t: $\lambda_t > 0$
$\boldsymbol{\mu}^{[\bullet]_t}$	Mean returns at node $[\bullet]_t$: $\boldsymbol{\mu}^{[\bullet]_t} \in \mathbb{R}^N$
$\mu_n^{[\bullet]_t}$	Mean return for security n at node $[\bullet]_t$: $-\infty < \mu_n^{[\bullet]_t} < \infty$
$\boldsymbol{\pi}^{(0)[\bullet]_t}$	Duals: budget constraint/portfolio upper bounds at node $[\bullet]_t$: $\boldsymbol{\pi}^{(0)[\bullet]_t} \in \mathbb{R}^{1+N}$
$\pi_0^{(0)[\bullet]_t}$	Dual: budget constraint at node $[\bullet]_t$: $\pi_0^{(0)[\bullet]_t} \geq 0$
$\pi_n^{(0)[\bullet]_t}$	Dual: security n portfolio variable upper bound at node $[\bullet]_t$: $\pi_n^{(0)[\bullet]_t} \geq 0$
$\boldsymbol{\pi}^{(1)[\bullet]_t}$	Duals: slippage constraints/$X_{kn}^{(1)[\bullet]_t}$ upper bounds at node $[\bullet]_t$: $\boldsymbol{\pi}^{(1)[\bullet]_t} \in \mathbb{R}^{(K+1)N}$
$\pi_n^{(1)[\bullet]_t}$	Dual: security n buy slippage constraint at node $[\bullet]_t$: $\pi_n^{(1)[\bullet]_t} \geq 0$
$\pi_{N+n}^{(1)[\bullet]_t}$	Dual: security n sell slippage constraint at node $[\bullet]_t$: $\pi_{N+n}^{(1)[\bullet]_t} \geq 0$
$\pi_{2N+(n-1)(K-1)+k}^{(1)[\bullet]_t}$	Dual: $X_{kn}^{(1)[\bullet]_t}$ upper bound ($1 \leq k \leq K-1$) at node $[\bullet]_t$: $\pi_{2N+(n+1)(K-1)+k}^{(1)[\bullet]_t} \geq 0$
$\boldsymbol{\pi}^{(2)[\bullet]_t}$	Duals: deviation constraints/$X_{ql_t}^{(2)[\bullet]_t}$ upper bounds at node $[\bullet]_t$: $\boldsymbol{\pi}^{(2)[\bullet]_t} \in \mathbb{R}^{QL_t}$
$\pi_{l_t}^{(2)[\bullet]_t}$	Dual: outcome l_t deviation constraint at node $[\bullet]_t$: $\pi_{l_t}^{(2)[\bullet]_t} \geq 0$
$\pi_{L_t+(l_t-1)(Q-1)+q}^{(1)[\bullet]_t}$	Dual: $X_{ql_t}^{(2)[\bullet]_t}$ upper bound ($1 \leq q \leq Q-1$) at node $[\bullet]_t$: $\pi_{L_t+(l_t-1)(Q-1)+q}^{(1)[\bullet]_t} \geq 0$
ϱ_t	Capital discount factor for period t: $0 < \varrho_t \leq 1$, $\varrho_1 = 1$
$\hat{\varrho}_t$	Compound capital discount factor for period t: $\hat{\varrho}_t = \prod_{j=1}^{t} \varrho_j$
$\boldsymbol{\sigma}^{[\bullet]_t}$	Standard deviations on returns at node $[\bullet]_t$: $\boldsymbol{\sigma}^{[\bullet]_t} \in \mathbb{R}^N$
$\sigma_n^{[\bullet]_t}$	Standard deviation on return for security n at node $[\bullet]_t$: $0 < \sigma_n^{[\bullet]_t} < \infty$
$\boldsymbol{\Phi}$	Matrix of slippage PWL slopes for all securities: $\boldsymbol{\Phi} \in \mathbb{R}^{K \times N}$
Φ_{kn}	Slippage PWL slope for security n in segment k: $\Phi_{1n} < \cdots < \Phi_{kn} < \ldots < \Phi_{Kn}$
φ_q	Downside deviation PWL upper break point in segment q: $0 = \varphi_0 < \varphi_1 < \cdots < \varphi_q < \cdots < \varphi_Q = \infty$
$\boldsymbol{\Psi}$	Matrix of slippage PWL break points for all securities: $\boldsymbol{\Psi} \in \mathbb{R}^{(1+K) \times N}$
Ψ_{kn}	Slippage PWL upper break point for security n in segment k: $0 = \Psi_{0n} < \Psi_{1n} < \cdots < \Psi_{kn} < \cdots < \Psi_{Kn} = \infty$
$\pounds_t$	Length in days for period t, $1 \leq t \leq T$
$[\bullet]_t$	Path vector to node $[\bullet]_t = \boxed{\tilde{l}_1, \ldots, \tilde{l}_{t-1}}$
(t, h_t)	Period-index label for node $[\bullet]_t$: see equations [2.6] and [2.7] to relate $[\bullet]_t$ and (t, h_t)

$[\bullet]_t$, $1 \le t \le T$, the primal grand LP formulation for a general problem is:

$$z = \max \sum_{t=1}^{T} \hat{\varrho}_t \left[\sum_{h_t=1}^{H_t} \hat{p}^{[\bullet]_t} \left(\sum_{n=1}^{N} \mu_n^{[\bullet]_t} y_n^{[\bullet]_t} - \sum_{n=1}^{N} \sum_{k=1}^{K} \Phi_{kn} X_{kn}^{(1)[\bullet]_t} - \lambda_t \sum_{l_t=1}^{L_t} p_{l_t}^{[\bullet]_t} \sum_{q=1}^{Q} \gamma_q X_{ql_t}^{(2)[\bullet]_t} \right) \right] \qquad [5.17]$$

s.t. constraints at the single first period node,

$$\begin{aligned}
\sum_{n=1}^{N} \sigma_n^{[]} y_n^{[]} &\le \beta_1, \\
y_n^{[]} - \sum_{k=1}^{K} X_{kn}^{(1)[]} &\le y_n^{(0)}, \quad n = 1, \ldots, N, \\
y_n^{[]} + \sum_{k=1}^{K} X_{kn}^{(1)[]} &\ge y_n^{(0)}, \quad n = 1, \ldots, N, \\
\sum_{n=1}^{N} \left(\mu_n^{[]} - R_{nl_1}^{[]} \right) y_n^{[]} - \sum_{q=1}^{Q} X_{ql_1}^{(2)[]} &\le 0, \quad l_1 = 1, \ldots, L_1,
\end{aligned}$$

constraints at nodes $[\bullet]_t = (t, h_t)$, $h_t = 1, \ldots, H_t, t = 2, \ldots, T$,

$$\begin{aligned}
\sum_{n=1}^{N} \sigma_n^{[\bullet]_t} y_n^{[\bullet]_t} &\le \beta_t, \\
-y_n^{[\bullet]_{t-1}} + y_n^{[\bullet]_t} - \sum_{k=1}^{K} X_{kn}^{(1)[\bullet]_t} &\le 0, \quad n = 1, \ldots, N, \\
-y_n^{[\bullet]_{t-1}} + y_n^{[\bullet]_t} + \sum_{k=1}^{K} X_{kn}^{(1)[\bullet]_t} &\ge 0, \quad n = 1, \ldots, N, \\
\sum_{n=1}^{N} \left(\mu_n^{[\bullet]_t} - R_{nl_t}^{[\bullet]_t} \right) y_n^{[\bullet]_t} - \sum_{q=1}^{Q} X_{ql_t}^{(2)[\bullet]_t} &\le 0, \quad l_t = 1, \ldots, L_t,
\end{aligned}$$

and bounds at nodes $[\bullet]_t = (t, h_t)$, $h_t = 1, \ldots, H_t, t = 1, \ldots, T$,

$$Y_{nt}^{\min} \le y_n^{[\bullet]_t} \le Y_{nt}^{\max}, \qquad n = 1, \ldots, N,$$

$$0 \le X_{kn}^{(1)} \le \Psi_{kn} - \Psi_{(k-1)n}, \quad k = 1, \ldots, K, n = 1, \ldots, N,$$

$$0 \le X_{ql_t}^{(2)} \le \varphi_q - \varphi_{q-1}, \qquad q = 1, \ldots, Q, l_t = 1, \ldots, L_t.$$

The following simplifying notation is introduced to make the problems more manageable. First, the variable substitution:

$$\mathbf{x}^{(0)[\bullet]_t} = \mathbf{y}^{[\bullet]_t} - \mathbf{Y}_{\bullet t}^{\min}, \; h_t = 1, \ldots, H_t, t = 1, \ldots, T, \qquad [5.18]$$

is made for the portfolio variables at all nodes. Variables $\mathbf{x}^{(0)[\bullet]_t}$ are referred to as *translated* portfolio variables while variables $\mathbf{y}^{[\bullet]_t}$ are called *non-translated* portfolio, or simply portfolio, variables. Introduction of translated portfolio variables results in lower bounds of zero on these variables and simplifies construction of the dual problem. The portfolio variables' substitutions also induce the additive objective function constant:

$$z_{\text{con}} = \sum_{t=1}^{T} \hat{\varrho}_t \left[\sum_{h_t=1}^{H_t} \hat{p}^{[\bullet]_t} \left(\sum_{n=1}^{N} \mu_n^{[\bullet]_t} Y_{nt}^{\min} \right) \right] = \sum_{t=1}^{T} \hat{\varrho}_t \left[\sum_{h_t=1}^{H_t} \hat{p}^{[\bullet]_t} \left(\boldsymbol{\mu}'^{[\bullet]_t} \mathbf{Y}_{\bullet t}^{\min} \right) \right],$$

and the following changes to the right-hand-sides of the indicated constraints at all nodes $[\bullet]_t = (t, h_t)$, $h_t = 1, \ldots, H_t$, $t = 1, \ldots, T$:

$$\hat{\beta}^{[\bullet]_t} = \beta_t - \boldsymbol{\sigma}'^{[\bullet]_t} \mathbf{Y}_{\bullet t}^{\min}, \qquad \text{budget constraints,}$$

$$\triangle \mathbf{Y}_{\bullet t}^{\min} = \begin{cases} \mathbf{y}^{(0)} - \mathbf{Y}_{\bullet 1}^{\min}, & \text{if } t = 1 \\ \mathbf{Y}_{\bullet (t-1)}^{\min} - \mathbf{Y}_{\bullet t}^{\min}, & \text{if } t \in \{2, \ldots, T\} \end{cases}, \qquad \text{slippage constraints,}$$

$$\hat{d}_{l_t}^{[\bullet]_t} = \sum_{n=1}^{N} \left(R_{nl_t}^{[\bullet]_t} - \mu_n^{[\bullet]_t} \right) Y_{nt}^{\min}, l_t = 1, \ldots, L_t, \qquad \text{deviation constraints,}$$

$$\triangle \mathbf{Y}_{\bullet t}^{\max} = \mathbf{Y}_{\bullet t}^{\max} - \mathbf{Y}_{\bullet t}^{\min}, \qquad \text{upper bounds on } \mathbf{x}^{(0)[\bullet]_t}.$$

Computations are reduced by defining the following variables representing the upper bounds on the translated portfolio variables and the PWL slippage and deviation variables at all nodes:

$$\mathbf{x}^{(0)[\bullet]_t} \leq \triangle \mathbf{Y}_{\bullet t}^{\max} = \mathbf{Y}_{\bullet t}^{\max} - \mathbf{Y}_{\bullet t}^{\min},$$

$$X_{kn}^{(1)[\bullet]_t} \leq \triangle \Psi_{kn} = \Psi_{kn} - \Psi_{(k-1)n}, \quad k = 1, \ldots, K, \; n = 1, \ldots, N,$$

$$X_{ql_t}^{(2)[\bullet]_t} \leq \triangle \varphi_q = \varphi_q - \varphi_{q-1}, \qquad q = 1, \ldots, Q, l_t = 1, \ldots, L_t.$$

Finally, the imaginary period zero portfolio variable:

$$\mathbf{x}^{(0)[\bullet]_0} = \mathbf{0},$$

simplifies construction of the problem at the single first period node. Table 5.2 summarizes this new notation and is provided as a convenient reference.

Primal problem [5.17] is then reformulated as problem **PMPGLP**:

$$\hat{z} = \max \sum_{t=1}^{T} \hat{\varrho}_t \left[\sum_{h_t=1}^{H_t} \hat{p}^{[\bullet]_t} \left(\sum_{n=1}^{N} \mu_n^{[\bullet]_t} x_n^{(0)[\bullet]_t} - \sum_{n=1}^{N} \sum_{k=1}^{K} \Phi_{kn} X_{kn}^{(1)[\bullet]_t} - \lambda_t \sum_{l_t=1}^{L_t} p_{l_t}^{[\bullet]_t} \sum_{q=1}^{Q} \gamma_q X_{ql_t}^{(2)[\bullet]_t} \right) \right] \qquad [5.19]$$

s.t. constraints at nodes $[\bullet]_t = (t, h_t)\,,\ h_t = 1, \ldots, H_t, t = 1, \ldots, T,$

$$\begin{aligned}
\sum_{n=1}^{N} \sigma_n^{[\bullet]_t} x_n^{(0)[\bullet]_t} &\leq \hat{\beta}^{[\bullet]_t}, \\
-x_n^{(0)[\bullet]_{t-1}} + x_n^{(0)[\bullet]_t} - \sum_{k=1}^{K} X_{kn}^{(1)[\bullet]_t} &\leq \Delta Y_{nt}^{\min}, \quad n = 1, \ldots, N, \\
x_n^{(0)[\bullet]_{t-1}} - x_n^{(0)[\bullet]_t} - \sum_{k=1}^{K} X_{kn}^{(1)[\bullet]_t} &\leq -\Delta Y_{nt}^{\min}, \quad n = 1, \ldots, N, \\
\sum_{n=1}^{N} \left(\mu_n^{[\bullet]_t} - R_{nl_t}^{[\bullet]_t} \right) x_n^{(0)[\bullet]_t} - \sum_{q=1}^{Q} X_{ql_t}^{(2)[\bullet]_t} &\leq \hat{d}_{l_t}^{[\bullet]_t}, \quad l_t = 1, \ldots, L_t,
\end{aligned}$$

and bounds at nodes $[\bullet]_t = (t, h_t)\,,\ h_t = 1, \ldots, H_t, t = 1, \ldots, T,$

Table 5.2: Model **MIMPSLP** Simplifying Notation

Notation	Description[a]	Reason
$\mathbf{x}^{(0)[\bullet]_t} = \mathbf{y}^{[\bullet]_t} - \mathbf{Y}^{\min}_{\bullet t}$	portfolio variables' substitution	induce lower bounds of zero
$z_{\text{con}} = \sum_{t=1}^{T} \hat{\varrho}_t \left[\sum_{h_t=1}^{H_t} \hat{p}^{[\bullet]_t} \left(\mu'^{[\bullet]_t} \mathbf{Y}^{\min}_{\bullet t} \right) \right]$	objective function constant	variable substitution
$\hat{\beta}^{[\bullet]_t} = \beta_t - \boldsymbol{\sigma}'^{[\bullet]_t} \mathbf{Y}^{\min}_{\bullet t}$	RHS of budget constraints	variable substitution
$\triangle \mathbf{Y}^{\max}_{\bullet t} = \mathbf{Y}^{\max}_{\bullet t} - \mathbf{Y}^{\min}_{\bullet t}$	upper bounds on $\mathbf{x}^{(0)[\bullet]_t}$	variable substitution
$\triangle \mathbf{Y}^{\min}_{\bullet t} = \begin{cases} \mathbf{y}^{(0)} - \mathbf{Y}^{\min}_{\bullet 1}, & t = 1 \\ \mathbf{Y}^{\min}_{\bullet (t-1)} - \mathbf{Y}^{\min}_{\bullet t}, & t = 2, \ldots, T \end{cases}$	RHS of slippage constraints	variable substitution
$\hat{d}^{[\bullet]_t}_{l_t} = \sum_{n=1}^{N} \left(R^{[\bullet]_t}_{nl_t} - \mu^{[\bullet]_t}_{n} \right) Y^{\min}_{nt}, l_t = 1, \ldots, L_t$	RHS of deviation constraints	variable substitution
$\triangle \Psi_{kn} = \Psi_{kn} - \Psi_{(k-1)n}, k = 1, \ldots, K$	upper bounds on slippage variables	computational convenience
$\triangle \varphi_q = \varphi_q - \varphi_{q-1}, q = 1, \ldots, Q$	upper bounds on deviation variables	computational convenience
$\mathbf{x}^{(0)[\bullet]_0} = \mathbf{0}$	imaginary period 0 portfolio variables	problem construction

[a] RHS means right-hand-side

$$
\begin{aligned}
& x_n^{(0)[\bullet]_t} \leq \Delta Y_{nt}^{\max}, && n = 1, \ldots, N, \\
& X_{kn}^{(1)[\bullet]_t} \leq \Delta \Psi_{kn}, && k = 1, \ldots, K-1, \quad n = 1, \ldots, N, \\
& X_{ql_t}^{(2)[\bullet]_t} \leq \Delta \varphi_q, && q = 1, \ldots, Q-1, \quad l_t = 1, \ldots, L_t, \\
& x_n^{(0)[\bullet]_t} \geq 0, && n = 1, \ldots, N, \\
& X_{kn}^{(1)[\bullet]_t} \geq 0, && k = 1, \ldots, K, \quad n = 1, \ldots, N, \\
& X_{ql_t}^{(2)[\bullet]_t} \geq 0, && q = 1, \ldots, Q, \quad l_t = 1, \ldots, L_t.
\end{aligned}
$$

Note that the inequality on the slippage buy constraints (third set of constraints) has been reversed from the original formulation in order to put the problem in canonical form (e.g., see Bazaraa, Jarvis, and Sherali [6, page 5 and Table 1.1]). Also note that upper bounds are not included at any node for slippage variables $X_{Kn}^{(1)[\bullet]_t}$, $n = 1, \ldots, N$, or for deviation variables $X_{Ql_t}^{(2)[\bullet]_t}$, $l_t = 1, \ldots, L_t$. These bounds need not be explicitly listed since $\Delta \varphi_Q = \infty$ and $\Delta \Psi_{Kn} = \infty$ for $n = 1, \ldots, N$. Furthermore, the portfolio variables' substitutions (equations [5.18]) require that the objective function constant z_{con} defined above (or see the second row of Table 5.2) be added to the above result to obtain the true problem value:

$$z = \hat{z} + z_{\text{con}}. \qquad [5.20]$$

Dual Formulation

Three sets of dual multipliers are defined for each node in order to construct the dual problem to problem [5.19]. These definitions are valid at each node $[\bullet]_t = (t, h_t)$, $h_t = 1, \ldots, H_t$, $t = 1, \ldots, T$. First, let $\pi^{(0)[\bullet]_t} \in \mathbb{R}^{N+1}$, referred to as the dual *budget*

variables, be the vector of multipliers for the budget constraint and the upper bounding constraints on the translated portfolio variables with vector indices corresponding to the applicable primal constraints:

$$\begin{aligned} &\sum_{n=1}^{N} \sigma_n^{[\bullet]_t} x_n^{(0)[\bullet]_t} \leq \hat{\beta}^{[\bullet]_t}: && \pi_0^{(0)[\bullet]_t}, \\ &x_n^{(0)[\bullet]_t} \leq \Delta Y_{nt}^{\max},\ n = 1, \ldots, N: && \pi_n^{(0)[\bullet]_t},\ n = 1, \ldots, N. \end{aligned} \qquad [5.21]$$

Note that no dual budget variable, $\pi_n^{(0)[\bullet]_t}$, is required for any primal portfolio variable upper bounding constraint that has $\Delta Y_{nt}^{\max} = \infty$ since any such dual variable would always have a value of zero. Duals to the slippage constraints and upper bounding constraints on the slippage variables are called the dual *slippage variables* and are denoted by $\pi^{(1)[\bullet]_t} \in \mathbb{R}^{(K+1)N}$ with vector indices:

$$-x_n^{(0)[\bullet]_{t-1}} + x_n^{(0)[\bullet]_t} - \sum_{k=1}^{K} X_{kn}^{(1)[\bullet]_t} \leq \Delta Y_{nt}^{\min},\ n = 1, \ldots, N: \quad \pi_j^{(1)[\bullet]_t},\ j = 1, \ldots, N,$$

$$x_n^{(0)[\bullet]_{t-1}} - x_n^{(0)[\bullet]_t} - \sum_{k=1}^{K} X_{kn}^{(1)[\bullet]_t} \leq -\Delta Y_{nt}^{\min},\ n = 1, \ldots, N: \quad \pi_j^{(1)[\bullet]_t},\ j = N+1, \ldots, 2N,$$

$$X_{kn}^{(1)[\bullet]_t} \leq \Delta \Psi_{kn},\ k = 1, \ldots, K-1,\ n = 1, \ldots, N: \qquad \pi_j^{(1)[\bullet]_t},\ j = 2N+1, \ldots, (K+1)N. \qquad [5.22]$$

Finally, the duals to the deviation constraints and upper bounding constraints on the deviation variables are called the dual *deviation variables* and are denoted by $\pi^{(2)[\bullet]_t} \in \mathbb{R}^{QL_t}$ with vector indices:

$$\sum_{n=1}^{N} \left(\mu_n^{[\bullet]_t} - R_{nl_t}^{[\bullet]_t} \right) x_n^{(0)[\bullet]_t} - \sum_{q=1}^{Q} X_{ql_t}^{(2)[\bullet]_t} \leq \hat{d}_{l_t}^{[\bullet]_t},\ l_t = 1, \ldots, L_t: \quad \pi_j^{(2)[\bullet]_t},\ j = 1, \ldots, L_t,$$

$$X_{ql_t}^{(2)[\bullet]_t} \leq \Delta \varphi_q,\ q = 1, \ldots, Q-1,\ l_t = 1, \ldots, L_t: \qquad \pi_j^{(2)[\bullet]_t},\ j = L_t+1, \ldots, QL_t. \qquad [5.23]$$

Slippage duals, $\pi^{(1)[\bullet]_{T+1}} = 0$, are also defined for all fictitious nodes in an imaginary period $T+1$ for problem construction purposes only. Then the dual to problem **PMPGLP** [5.19] is referred to as problem **DMPGLP** and is:

$$\hat{z} = \min \sum_{t=1}^{T} \sum_{h_t=1}^{H_t} \left[\hat{\beta}^{[\bullet]_t} \pi_0^{(0)[\bullet]_t} + \sum_{n=1}^{N} \triangle Y_{nt}^{\max} \pi_n^{(0)[\bullet]_t} \right. \qquad [5.24]$$
$$+ \sum_{n=1}^{N} \triangle Y_{nt}^{\min} \left(\pi_n^{(1)[\bullet]_t} - \pi_{N+n}^{(1)[\bullet]_t} \right) + \sum_{n=1}^{N} \sum_{k=1}^{K-1} \triangle \Psi_{kn} \pi_{2N+(n-1)(K-1)+k}^{(1)[\bullet]_t}$$
$$\left. + \sum_{l_t=1}^{L_t} \hat{d}_{l_t}^{[\bullet]_t} \pi_{l_t}^{(2)[\bullet]_t} + \sum_{l_t=1}^{L_t} \sum_{q=1}^{Q-1} \triangle \varphi_q \pi_{L_t+(l_t-1)(Q-1)+q}^{(2)[\bullet]_t} \right]$$

s.t. constraints at nodes $[\bullet]_t = (t, h_t)$, $h_t = 1, \ldots, H_t$, $t = 1, \ldots, T$,

$$\sigma_n^{[\bullet]_t} \pi_0^{(0)[\bullet]_t} + \pi_n^{(0)[\bullet]_t} + \pi_n^{(1)[\bullet]_t} - \pi_{N+n}^{(1)[\bullet]_t} + \sum_{l_t=1}^{L_t} \left(\mu_n^{[\bullet]_t} - R_{nl_t}^{[\bullet]_t} \right) \pi_{l_t}^{(2)[\bullet]_t}$$
$$- \sum_{l_t=1}^{L_t} \left(\pi_n^{(1)[\bullet]_{t+1}} - \pi_{N+n}^{(1)[\bullet]_{t+1}} \right) \geq \hat{\varrho}_t \hat{p}^{[\bullet]_t} \mu_n^{[\bullet]_t}, \quad n = 1, \ldots, N,$$

$$-\pi_n^{(1)[\bullet]_t} - \pi_{N+n}^{(1)[\bullet]_t} + \pi_{2N+(n-1)(K-1)+k}^{(1)[\bullet]_t} \geq -\hat{\varrho}_t \hat{p}^{[\bullet]_t} \Phi_{kn}, \quad \begin{matrix} k = 1, \ldots, K-1, \\ n = 1, \ldots, N, \end{matrix}$$

$$-\pi_n^{(1)[\bullet]_t} - \pi_{N+n}^{(1)[\bullet]_t} \geq -\hat{\varrho}_t \hat{p}^{[\bullet]_t} \Phi_{Kn}, \quad n = 1, \ldots, N,$$

$$-\pi_{l_t}^{(2)[\bullet]_t} + \pi_{L_t+(l_t-1)(Q-1)+q}^{(2)[\bullet]_t} \geq -\hat{\varrho}_t \hat{p}^{[\bullet]_t} \lambda_t p_{l_t}^{[\bullet]_t} \gamma_q, \quad \begin{matrix} q = 1, \ldots, Q-1, \\ l_t = 1, \ldots, L_t, \end{matrix}$$

$$-\pi_{l_t}^{(2)[\bullet]_t} \geq -\hat{\varrho}_t \hat{p}^{[\bullet]_t} \lambda_t p_{l_t}^{[\bullet]_t} \gamma_Q, \quad l_t = 1, \ldots, L_t,$$

and lower bounds at nodes $[\bullet]_t = (t, h_t)$, $h_t = 1, \ldots, H_t$, $t = 1, \ldots, T$,

$$\pi_n^{(0)[\bullet]_t} \geq 0, \quad n = 0, \ldots, N,$$

$$\pi_j^{(1)[\bullet]_t} \geq 0, \quad j = 1, \ldots, (K+1)\,N,$$

$$\pi_j^{(2)[\bullet]_t} \geq 0, \quad j = 1, \ldots, QL_t.$$

The first set of N constraints at each node are known as the dual *composite constraints* because they incorporate all three types of dual variables. The next pair of constraint sets are referred to as the dual *slippage constraints* since they involve only the dual slippage variables. Finally, the last pair of constraint sets are called the dual *deviation constraints* since they involve only the dual deviation variables.

Note that similar to the primal problem **PMPGLP** [5.19], equation [5.20] must be applied to the objective function value of problem **DMPGLP** [5.24] in order to obtain the value for the original problem.

5.2 Model Problems in Array Notation

Problems **PMPGLP** [5.19] and **DMPGLP** [5.24] of model **MIMPSLP** can be put into the array notation defined in Chapter 2. Definitions for problem vectors and matrices are given below followed by the formulations in array notation. The following notational conventions are used to define problem arrays:

$\lfloor j \rfloor$: floor of j – largest integer less than or equal to j,

$[0]_{M \times N}$: M-by-N matrix of zeros, and

$\mathbf{A}\,(i_1{:}i_2, j_1{:}j_2)$**:** array colon notation – the submatrix of $\mathbf{A}$ containing elements in rows i_1 through i_2 and columns j_1 through j_2.

Also recall from Section 1.4 that $\mathbf{I}_M$ represents the M-by-M identity matrix. All arrays are defined relative to the primal formulation, problem **PMPGLP** [5.19], and the transposes of these arrays are used as appropriate in the dual problem **DMPGLP** [5.24]. Upper bounding constraints are included in the definitions for the right-hand-side vectors and the transition and recourse matrices.

5.2.1 Problem Vectors

Composite vectors $\mathbf{x}^{[\bullet]_t}$, $\mathbf{c}^{[\bullet]_t}$, and $\mathbf{b}^{[\bullet]_t}$ are used to represent the decision variables, cost coefficients, and right-hand-sides respectively for each node $[\bullet]_t$ in the decision tree. Vector $\mathbf{x}^{[\bullet]_t} \in \mathbb{R}^{(K+1)N+QL_t}$ is defined as

$$\mathbf{x}^{[\bullet]_t} = \begin{bmatrix} \mathbf{x}^{(0)[\bullet]_t} \\ \mathbf{x}^{(1)[\bullet]_t} \\ \mathbf{x}^{(2)[\bullet]_t} \end{bmatrix}, \qquad [5.25]$$

where $\mathbf{x}^{(0)[\bullet]_t} \in \mathbb{R}^N$ is defined by equation [5.18], $\mathbf{x}^{(1)[\bullet]_t} \in \mathbb{R}^{KN}$ such that

$$x_j^{(1)[\bullet]_t} = X_{k_j n_j}^{(1)[\bullet]_t} \text{ where } \begin{cases} n_j = \left\lfloor \frac{j-1}{K} \right\rfloor + 1 \\ k_j = j - (n_j - 1)\,K \end{cases} \text{ for } j = 1, \ldots, KN, \qquad [5.26]$$

and $\mathbf{x}^{(2)[\bullet]_t} \in \mathbb{R}^{QL_t}$ such that

$$x_j^{(2)[\bullet]_t} = X_{q_j l_j}^{(2)[\bullet]_t} \text{ where } \begin{cases} l_j = \left\lfloor \frac{j-1}{Q} \right\rfloor + 1 \\ q_j = j - (l_j - 1)\,Q \end{cases} \text{ for } j = 1, \ldots, QL_t. \qquad [5.27]$$

Cost coefficient vector $\mathbf{c}^{[\bullet]_t} \in \mathbb{R}^{(K+1)N+QL_t}$ is defined as

$$\mathbf{c}^{[\bullet]_t} = \begin{bmatrix} \mathbf{c}^{(0)[\bullet]_t} \\ \mathbf{c}^{(1)[\bullet]_t} \\ \mathbf{c}^{(2)[\bullet]_t} \end{bmatrix}, \qquad [5.28]$$

where $\mathbf{c}^{(0)[\bullet]_t} \in \mathbb{R}^N$ such that

$$\mathbf{c}^{(0)[\bullet]_t} = \hat{\varrho}_t \hat{p}^{[\bullet]_t} \boldsymbol{\mu}^{[\bullet]_t}, \tag{5.29}$$

$\mathbf{c}^{(1)[\bullet]_t} \in \mathbb{R}^{KN}$ such that

$$c_j^{(1)[\bullet]_t} = \hat{\varrho}_t \hat{p}^{[\bullet]_t} \Phi_{k_j n_j}^{(1)[\bullet]_t} \text{ where } \begin{cases} n_j = \left\lfloor \frac{j-1}{K} \right\rfloor + 1 \\ k_j = j - (n_j - 1) K \end{cases} \text{ for } j = 1, \ldots, KN, \tag{5.30}$$

and $\mathbf{c}^{(2)[\bullet]_t} \in \mathbb{R}^{QL_t}$ such that

$$c_j^{(2)[\bullet]_t} = \hat{\varrho}_t \hat{p}^{[\bullet]_t} \lambda_t p_{l_j}^{[\bullet]_t} \gamma_{q_j} \text{ where } \begin{cases} l_j = \left\lfloor \frac{j-1}{Q} \right\rfloor + 1 \\ q_j = j - (l_j - 1) Q \end{cases} \text{ for } j = 1, \ldots, QL_t. \tag{5.31}$$

Right-hand-side vector $\mathbf{b}^{[\bullet]_t} \in \mathbb{R}^{1+(K+2)N+QL_t}$ is defined as

$$\mathbf{b}^{[\bullet]_t} = \begin{bmatrix} \mathbf{b}^{(0)[\bullet]_t} \\ \mathbf{b}^{(1)[\bullet]_t} \\ \mathbf{b}^{(2)[\bullet]_t} \end{bmatrix}, \tag{5.32}$$

where $\mathbf{b}^{(0)[\bullet]_t} \in \mathbb{R}^{1+N}$ such that

$$b_n^{(0)[\bullet]_t} = \begin{cases} \hat{\beta}^{[\bullet]_t}, & n = 0 \\ \Delta Y_{nt}^{\max}, & n = 1, \ldots, N, \end{cases} \tag{5.33}$$

$\mathbf{b}^{(1)[\bullet]_t} \in \mathbb{R}^{(K+1)N}$ such that

$$b_j^{(1)[\bullet]_t} = \begin{cases} \Delta Y_{jt}^{\min}, & j = 1, \ldots, N, \\ -\Delta Y_{jt}^{\min}, & j = N+1, \ldots, 2N, \\ & j = 2N+1, \ldots, (K+1)N, \\ \Psi_{k_j n_j}, & \text{where } \begin{cases} n_j = \left\lfloor \frac{j-(2N+1)}{K-1} \right\rfloor + 1 \\ k_j = j - 2N - (n_j - 1)(K-1) \end{cases} \end{cases}, \tag{5.34}$$

and $\mathbf{b}^{(2)[\bullet]_t} \in \mathbb{R}^{QL_t}$ such that

$$b_j^{(2)[\bullet]_t} = \begin{cases} \hat{d}_j^{[\bullet]_t}, & j = 1, \ldots, L_t, \\ & j = L_t + 1, \ldots, QL_t, \\ \Delta \varphi_{q_j}, & \text{where } q_j = j - L_t - \left\lfloor \frac{j-(L_t+1)}{Q-1} \right\rfloor (Q-1). \end{cases} \tag{5.35}$$

5.2.2 Problem Matrices

Composite matrices $\mathbf{B}^{[\bullet]_t}$ and $\mathbf{W}^{[\bullet]_t}$ are used to represent the transition matrices and recourse matrices respectively. Transition matrix

$$\mathbf{B}^{[\bullet]_t} \in \mathbb{R}^{[1+(K+2)N+QL_t]\times[(K+1)N+QL_{t-1}]}$$

is defined for each node $[\bullet]_t$ in the decision tree in periods $t = 1, \ldots, T$ and in the imaginary period $t = T + 1$ as

$$\mathbf{B}^{[\bullet]_t} = \begin{bmatrix} [0]_{(1+N)\times N} & [0]_{(1+N)\times KN} & [0]_{(1+N)\times QL_{t-1}} \\ \mathbf{B}^{(1,0)[\bullet]_t} & [0]_{(K+1)N\times KN} & [0]_{(K+1)N\times QL_{t-1}} \\ [0]_{QL_t\times N} & [0]_{QL_t\times KN} & [0]_{QL_t\times QL_{t-1}} \end{bmatrix}, \qquad [5.36]$$

where $\mathbf{B}^{(1,0)[\bullet]_t} \in \mathbb{R}^{(K+1)N\times N}$ such that

$$\left.\begin{aligned} \mathbf{B}^{(1,0)[\bullet]_t}(1{:}N, 1{:}N) &= -\mathbf{I}_N \\ \mathbf{B}^{(1,0)[\bullet]_t}(N+1{:}2N, 1{:}N) &= \mathbf{I}_N \\ \mathbf{B}^{(1,0)[\bullet]_t}(2N+1{:}(K+1)N, 1{:}N) &= [0]_{(K-1)N\times N} \end{aligned}\right\} \text{ if } 2 \le t \le T, \qquad [5.37]$$

$$\left.\mathbf{B}^{(1,0)[]} = \mathbf{B}^{(1,0)[\bullet]_{T+1}} = [0]_{(K+1)N\times N}\right\} \text{ if } t = 1 \text{ or } T+1$$

Note that $\mathbf{B}^{(1,0)[]} = \mathbf{B}^{(1,0)[\bullet]_{T+1}} = [0]_{(K+1)N\times N}$ are defined solely for the purpose of simplifying the problem construction procedure.

Recourse matrix

$$\mathbf{W}^{[\bullet]_t} \in \mathbb{R}^{[1+(K+2)N+QL_t]\times[(K+1)N+QL_t]}$$

is defined for each node $[\bullet]_t$ in the decision tree in periods $t = 1, \ldots, T$ as

$$\mathbf{W}^{[\bullet]_t} = \begin{bmatrix} \mathbf{W}^{(0,0)[\bullet]_t} & [\mathbf{0}]_{(1+N)\times KN} & [\mathbf{0}]_{(1+N)\times QL_t} \\ \mathbf{W}^{(1,0)[\bullet]_t} & \mathbf{W}^{(1,1)[\bullet]_t} & [\mathbf{0}]_{(K+1)N\times QL_t} \\ \mathbf{W}^{(2,0)[\bullet]_t} & [\mathbf{0}]_{QL_t\times KN} & \mathbf{W}^{(2,2)[\bullet]_t} \end{bmatrix}, \qquad [5.38]$$

where the five non-zero submatrices of $\mathbf{W}^{[\bullet]_t}$ are defined as follows. Submatrix $\mathbf{W}^{(0,0)[\bullet]_t} \in \mathbb{R}^{(1+N)\times N}$ is defined as

$$\mathbf{W}_{0\bullet}^{(0,0)[\bullet]_t} = \boldsymbol{\sigma}'^{[\bullet]_t}, \qquad [5.39]$$

$$\mathbf{W}^{(0,0)[\bullet]_t}(1{:}N, 1{:}N) = \mathbf{I}_N,$$

submatrix $\mathbf{W}^{(1,0)[\bullet]_t} \in \mathbb{R}^{(K+1)N\times N}$ is defined as

$$\mathbf{W}^{(1,0)[\bullet]_t}(1{:}N, 1{:}N) = \mathbf{I}_N,$$

$$\mathbf{W}^{(1,0)[\bullet]_t}(N+1{:}2N, 1{:}N) = -\mathbf{I}_N, \qquad [5.40]$$

$$\mathbf{W}^{(1,0)[\bullet]_t}(2N+1{:}(K+1)N, 1{:}N) = [\mathbf{0}]_{(K-1)N\times N},$$

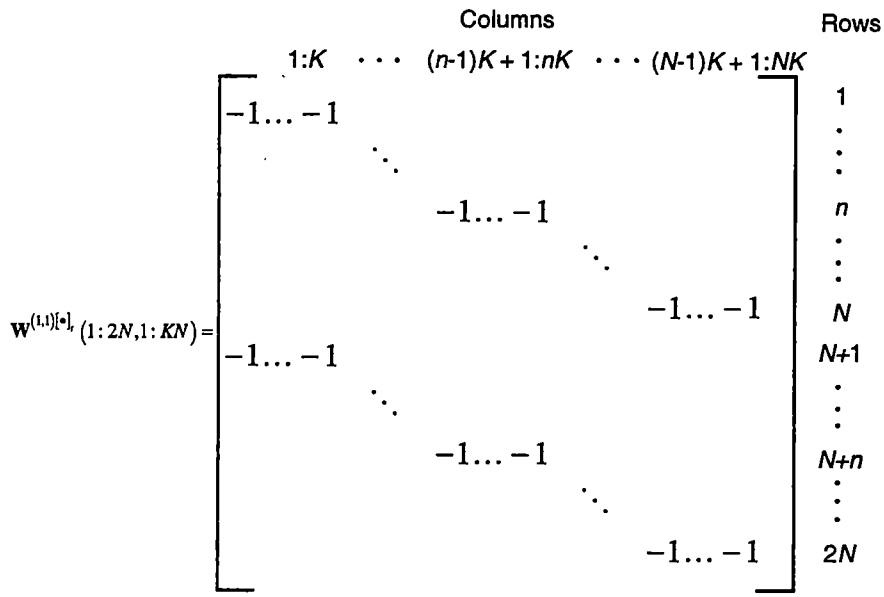
and submatrix $\mathbf{W}^{(2,0)[\bullet]_t} \in \mathbb{R}^{QL_t\times N}$ is defined as

$$\mathbf{W}_{l_t\bullet}^{(2,0)[\bullet]_t} = \left(\boldsymbol{\mu}^{[\bullet]_t} - \mathbf{R}_{\bullet l_t}^{[\bullet]_t}\right)', \; l_t = 1, \ldots, L_t, \qquad [5.41]$$

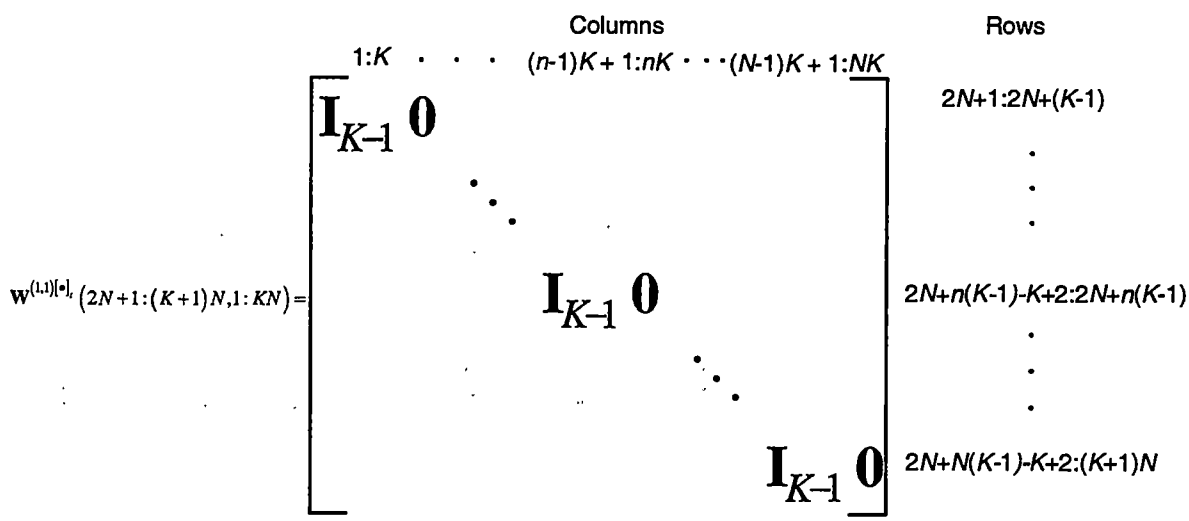
$$\mathbf{W}^{(2,0)[\bullet]_t}(L_t+1{:}QL_t, 1{:}N) = [\mathbf{0}]_{(Q-1)L_t\times N}.$$

The lower two diagonal submatrices $\mathbf{W}^{(1,1)[\bullet]_t} \in \mathbb{R}^{(K+1)N\times KN}$ and $\mathbf{W}^{(2,2)[\bullet]_t} \in \mathbb{R}^{QL_t\times QL_t}$ are defined in Figures 5.3 and 5.4 respectively. Note that the first period technology matrix is denoted by $\mathbf{W}^{[]}$ and not by $\mathbf{A}$ as in Chapters 2 and 4.

Upper Submatrix



Lower Submatrix

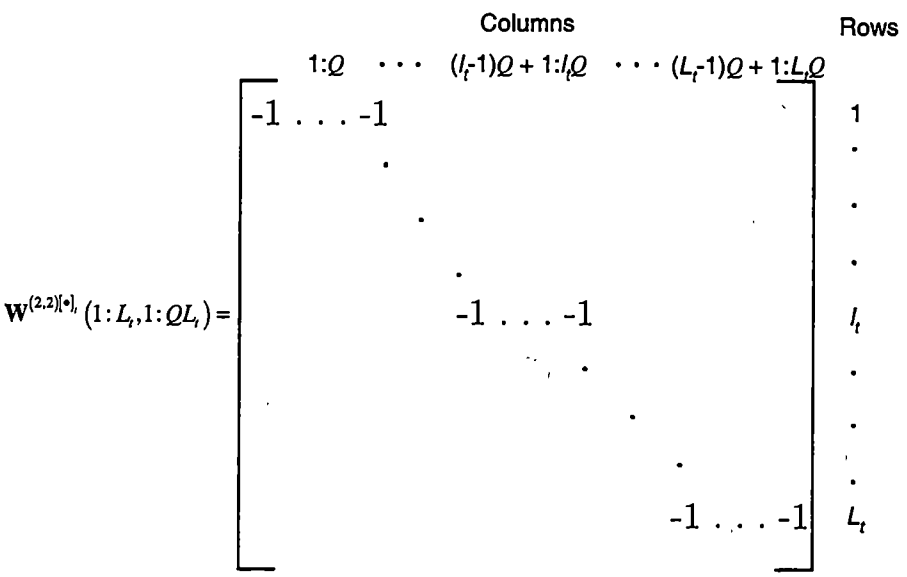


NOTES

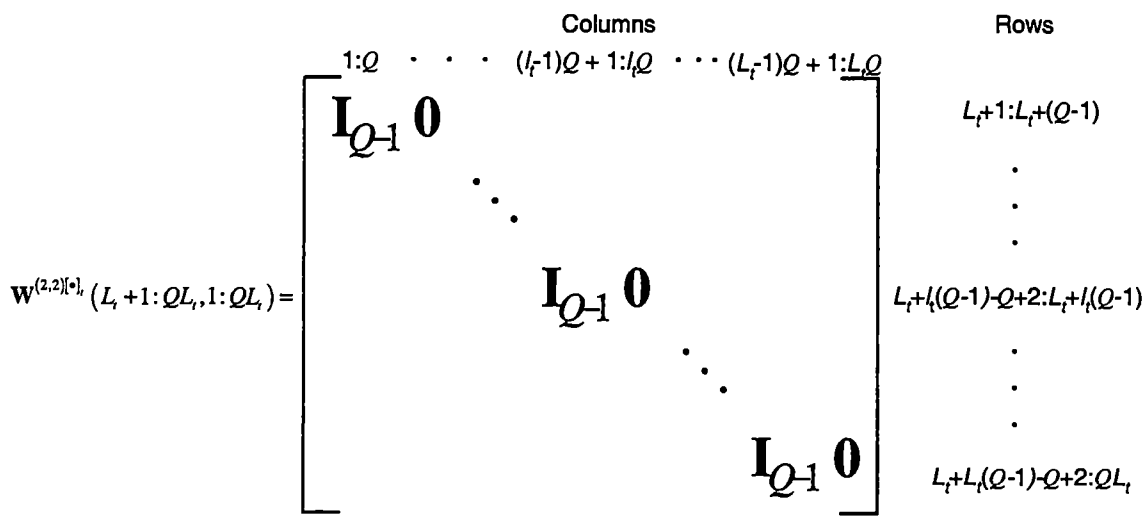
1. Blank portions of both submatrices filled with zeros
2. $\mathbf{I}_{K-1}$ represents the $(K-1)$-by-$(K-1)$ identity matrix
3. **0** represents a column vector of $(K-1)$ zeros

Figure 5.3: Recourse Submatrix $\mathbf{W}^{(1,1)[\bullet]_t}$

Upper Submatrix



Lower Submatrix



NOTES

1. Blank portions of both submatrices filled with zeros
2. $\mathbf{I}_{Q-1}$ represents the $(Q-1)$-by-$(Q-1)$ identity matrix
3. $\mathbf{0}$ represents a column vector of $(Q-1)$ zeros

Figure 5.4: Recourse Submatrix $\mathbf{W}^{(2,2)[\bullet]_t}$

5.2.3 Array Formulations

Problem **PMPGLP** [5.19] can then be written in the following formulation analogous to formulation [2.8] on page 35:

$$
\begin{aligned}
z = \max \quad & \sum_{t=1}^{T} \sum_{h_t=1}^{H_t} \mathbf{c}'^{[\bullet]_t} \mathbf{x}^{[\bullet]_t} \\
\text{s.t.} \quad & \mathbf{W}^{[]} \mathbf{x}^{[]} \leq \mathbf{b}^{[]}, \\
& \mathbf{B}^{[\bullet]_t} \mathbf{x}^{[\bullet]_{t-1}} + \mathbf{W}^{[\bullet]_t} \mathbf{x}^{[\bullet]_t} \leq \mathbf{b}^{[\bullet]_t}, \quad h_t = 1, \ldots, H_t, \quad t = 2, \ldots, T, \\
& \mathbf{x}^{[\bullet]_t} \geq \mathbf{0}, \quad h_t = 1, \ldots, H_t, \quad t = 1, \ldots, T.
\end{aligned}
\tag{5.42}
$$

Similarly, problem **DMPGLP** [5.24] can be written in the following formulation analogous to formulation [2.12] on page 39:

$$
\begin{aligned}
z = \min \quad & \sum_{t=1}^{T} \sum_{h_t=1}^{H_t} \mathbf{b}'^{[\bullet]_t} \boldsymbol{\pi}^{[\bullet]_t} \\
\text{s.t.} \quad & \mathbf{W}'^{[\bullet]_t} \boldsymbol{\pi}^{[\bullet]_t} + \sum_{l_t=1}^{L_t} \mathbf{B}'^{[\bullet]_{t+1}} \boldsymbol{\pi}^{[\bullet]_{t+1}} \geq \mathbf{c}^{[\bullet]_t}, \quad \begin{matrix} h_t = 1, \ldots, H_t, \\ t = 1, \ldots, T-1, \end{matrix} \\
& \mathbf{W}'^{[\bullet]_T} \boldsymbol{\pi}^{[\bullet]_T} \geq \mathbf{c}^{[\bullet]_T}, \quad h_T = 1, \ldots, H_T, \\
& \boldsymbol{\pi}^{[\bullet]_t} \geq \mathbf{0}, \quad \begin{matrix} h_t = 1, \ldots, H_t, \\ t = 1, \ldots, T, \end{matrix}
\end{aligned}
\tag{5.43}
$$

where $\boldsymbol{\pi}^{[\bullet]_t} \in \mathbb{R}^{1+(K+2)N+QL_t}$ such that

$$
\boldsymbol{\pi}^{[\bullet]_t} = \begin{bmatrix} \boldsymbol{\pi}^{(0)[\bullet]_t} \\ \boldsymbol{\pi}^{(1)[\bullet]_t} \\ \boldsymbol{\pi}^{(2)[\bullet]_t} \end{bmatrix}. \tag{5.44}
$$

5.3 Properties and Sizes of Model Problems

Problems **PMPGLP** [5.19] and **DMPGLP** [5.24] of model **MIMPSLP** possess several of the stochastic programming properties described in Chapters 2 and 4. Each problem has complete recourse, is block-separable, and models a sequential *decision-observation* stochastic process that terminates with an observation in each period of the planning horizon. These three properties are discussed below and additional properties of the model are introduced where appropriate. This section concludes with comments on the sizes of the two grand LP formulations.

5.3.1 Observation-Terminated Process

Model **MIMPSLP** mathematically describes a observation-terminated stochastic process as defined in Section 2.2.1. The end of each period in the planning horizon is simulated by the observation of the vector of random returns and the simultaneous determination of the downside deviation. Deviation variables $\mathbf{X}^{(2)[\bullet]_t}$ at each node $[\bullet]_t$ in the decision tree are reactive recourse variables since they model a automatic reaction of the system to a realized outcome. Translated portfolio variables $\mathbf{x}^{(0)[\bullet]_t}$ and slippage variables $\mathbf{X}^{(1)[\bullet]_t}$ at each node in the tree are considered to be discretionary recourse variables. The translated portfolio variables simulate investment decisions that must consider the fixed costs, or slippage, but must be made prior to the observation of the uncertain returns in the applicable period.

The observation-terminated market investment process requires two modifications to the notational conventions developed in Chapter 2 for stochastic linear programs that

model decision-terminated processes. One modification results from the presence of random returns in the first period of the planning horizon. The first period technology matrix is treated as a recourse matrix, denoted by $\mathbf{W}^{[]}$, since it contains stochastic elements whereas this matrix is fixed in a decision-terminated process. Termination of the final period with an observation induces the second modification which is to assign a number of outcomes, L_T, to each node in the terminal period. No outcomes are assigned to nodes in the final period of a decision-terminated process. Each of these modifications has been incorporated in the formulations developed in the previous section.

5.3.2 Complete Recourse

The primal budget constraints, $\boldsymbol{\sigma}'^{[\bullet]_t}\mathbf{x}^{(0)[\bullet]_t} \leq \hat{\beta}^{[\bullet]_t}$, insure that problem **PMPGLP** [5.19] is bounded since (see the third row of Table 5.2)

$$\hat{\beta}^{[\bullet]_t} = \beta_t - \boldsymbol{\sigma}'^{[\bullet]_t}\mathbf{Y}^{\min}_{\bullet t} < \infty$$

for all nodes $[\bullet]_t$ in periods $t = 1, \ldots, T$. In addition, if

$$\hat{\beta}^{[\bullet]_t} = \beta_t - \boldsymbol{\sigma}'^{[\bullet]_t}\mathbf{Y}^{\min}_{\bullet t} \geq 0 \iff \boldsymbol{\sigma}'^{[\bullet]_t}\mathbf{Y}^{\min}_{\bullet t} \leq \beta_t$$

at each node, then there exist translated portfolio vectors $\mathbf{x}^{(0)[\bullet]_t} \geq \mathbf{0}$ such that the primal budget constraints are satisfied. The above condition is checked as soon as the input data for the model is known and the program is terminated if the condition is violated at any node. Therefore, no solution procedure is initiated unless there exist values for the primal portfolio variables that satisfy the budget constraints.

Then note that given translated portfolio variables, there exist primal slippage variables $\mathbf{X}^{(1)[\bullet]_t} \geq 0$ and deviation variables $\mathbf{X}^{(2)[\bullet]_t} \geq 0$ at all nodes that satisfy the primal slippage constraints,

$$\left.\begin{aligned} -x_n^{(0)[\bullet]_{t-1}} + x_n^{(0)[\bullet]_t} - \sum_{k=1}^{K} X_{kn}^{(1)[\bullet]_t} \leq \triangle Y_{nt}^{\min}, \\ x_n^{(0)[\bullet]_{t-1}} - x_n^{(0)[\bullet]_t} - \sum_{k=1}^{K} X_{kn}^{(1)[\bullet]_t} \leq -\triangle Y_{nt}^{\min}, \\ X_{kn}^{(1)[\bullet]_t} \leq \triangle \Psi_{kn},\ k = 1, \ldots, K-1, \end{aligned}\right\} n = 1, \ldots, N,$$

and primal deviation constraints,

$$\left.\begin{aligned} \sum_{n=1}^{N} \left(\mu_n^{[\bullet]_t} - R_{nl_t}^{[\bullet]_t}\right) x_n^{(0)[\bullet]_t} - \sum_{q=1}^{Q} X_{ql_t}^{(2)[\bullet]_t} \leq \hat{d}_{l_t}^{[\bullet]_t}, \\ X_{ql_t}^{(2)[\bullet]_t} \leq \triangle \varphi_q,\ q = 1, \ldots, Q-1, \end{aligned}\right\} l_t = 1, \ldots, L_t,$$

respectively. Problem **PMPGLP** [5.19] is therefore bounded and considered to have complete recourse which also implies that the dual problem **DMPGLP** [5.24] has a feasible and bounded solution.

Given the problem array notation defined in Section 5.2 above, the only stochastic arrays are the recourse matrices $\mathbf{W}^{[\bullet]_t}$ [5.38], specifically the submatrices $\mathbf{W}^{(2,0)[\bullet]_t}$ [5.41]. Therefore, problems **PMPGLP** and **DMPGLP** do not have fixed recourse.

5.3.3 Block-Separable Recourse

Problems **PMPGLP** [5.19] and **DMPGLP** [5.24] of model **MIMPSLP** have the block-separable recourse property described in Section 2.4 starting on page 42. Block-separability is demonstrated by examining the left-hand-sides of the second set of constraints in the

array formulation of problem **PMPGLP** [5.42]. These left-hand-sides:

$$\mathbf{B}^{[\bullet]_t}\mathbf{x}^{[\bullet]_{t-1}} + \mathbf{W}^{[\bullet]_t}\mathbf{x}^{[\bullet]_t}$$

can be equivalently written with the subarray notation defined in Section 5.2. Transition matrix $\mathbf{B}^{[\bullet]_t}$ [5.36] is written in terms of submatrix $\mathbf{B}^{(1,0)[\bullet]_t}$ [5.37] and eight zero submatrices. Recourse matrix $\mathbf{W}^{[\bullet]_t}$ [5.38] is written in terms of four zero submatrices and the five non-zero submatrices defined by equations [5.39] through [5.41] and Figures 5.3 and 5.4. All dimensioning subscripts on the zero submatrices are omitted for brevity. Decision vector $\mathbf{x}^{[\bullet]_t}$ [5.25] is written in terms of the three subvectors $\mathbf{x}^{(0)[\bullet]_t}$ [5.18], $\mathbf{x}^{(1)[\bullet]_t}$ [5.26], and $\mathbf{x}^{(2)[\bullet]_t}$ [5.27]. The left-hand-sides in subarray notation are:

$$\begin{bmatrix} [0] & [0] & [0] \\ \mathbf{B}^{(1,0)[\bullet]_t} & [0] & [0] \\ [0] & [0] & [0] \end{bmatrix} \begin{bmatrix} \mathbf{x}^{(0)[\bullet]_{t-1}} \\ \mathbf{x}^{(1)[\bullet]_{t-1}} \\ \mathbf{x}^{(2)[\bullet]_{t-1}} \end{bmatrix} + \begin{bmatrix} \mathbf{W}^{(0,0)[\bullet]_t} & [0] & [0] \\ \mathbf{W}^{(1,0)[\bullet]_t} & \mathbf{W}^{(1,1)[\bullet]_t} & [0] \\ \mathbf{W}^{(2,0)[\bullet]_t} & [0] & \mathbf{W}^{(2,2)[\bullet]_t} \end{bmatrix} \begin{bmatrix} \mathbf{x}^{(0)[\bullet]_t} \\ \mathbf{x}^{(1)[\bullet]_t} \\ \mathbf{x}^{(2)[\bullet]_t} \end{bmatrix}. \quad [5.45]$$

Translated portfolio variables $\mathbf{x}^{(0)[\bullet]_t}$ are treated as aggregate level decision variables while primal deviation variables $\mathbf{x}^{(2)[\bullet]_t} \leftarrow \mathbf{X}^{(2)[\bullet]_t}$ are detailed level decision variables. Primal slippage variables $\mathbf{x}^{(1)[\bullet]_t} \leftarrow \mathbf{X}^{(1)[\bullet]_t}$ can be treated as either aggregate or detailed level decision variables. The best (fastest) results are obtained when the primal slippage variables are considered to be detailed level decision variables and this assignment is adopted hereinafter. The left-hand-sides given by relation [5.45] can be separated according to these assignments for aggregate and detailed level decision variables. Similar separations of cost

coefficient vectors $\mathbf{c}^{[\bullet]_t}$ [5.28] into the subvectors defined by equations [5.29] through [5.31] and right-hand-side vectors $\mathbf{b}^{[\bullet]_t}$ [5.32] into the subvectors defined by equations [5.33] through [5.35] are possible. These array separations allow problem **PMPGLP** [5.42] to be equivalently formulated as:

$$z = \max \sum_{t=1}^{T} \sum_{h_t=1}^{H_t} \mathbf{c}'^{(0)[\bullet]_t} \mathbf{x}^{(0)[\bullet]_t} + \sum_{t=1}^{T} \sum_{h_t=1}^{H_t} \left(\mathbf{c}'^{(1)[\bullet]_t} \mathbf{x}^{(1)[\bullet]_t} + \mathbf{c}'^{(2)[\bullet]_t} \mathbf{x}^{(2)[\bullet]_t} \right) \tag{5.46}$$

s.t. aggregate level constraints at nodes $[\bullet]_t = (t, h_t)$, $h_t = 1, \ldots, H_t$, $t = 1, \ldots, T$,

$$\begin{aligned} \mathbf{W}^{(0,0)[\bullet]_t} \mathbf{x}^{(0)[\bullet]_t} &\leq \mathbf{b}^{(0)[\bullet]_t}, \\ \mathbf{x}^{(0)[\bullet]_t} &\geq \mathbf{0}, \end{aligned}$$

and detailed level constraints at nodes $[\bullet]_t = (t, h_t)$, $h_t = 1, \ldots, H_t$, $t = 1, \ldots, T$,

$$\begin{aligned} \mathbf{B}^{(1,0)[\bullet]_t} \mathbf{x}^{(0)[\bullet]_{t-1}} + \mathbf{W}^{(1,0)[\bullet]_t} \mathbf{x}^{(0)[\bullet]_t} + \mathbf{W}^{(1,1)[\bullet]_t} \mathbf{x}^{(1)[\bullet]_t} &\leq \mathbf{b}^{(1)[\bullet]_t}, \\ \mathbf{W}^{(2,0)[\bullet]_t} \mathbf{x}^{(0)[\bullet]_t} + \mathbf{W}^{(2,2)[\bullet]_t} \mathbf{x}^{(2)[\bullet]_t} &\leq \mathbf{b}^{(2)[\bullet]_t}, \\ \mathbf{x}^{(1)[\bullet]_t} &\geq \mathbf{0}, \\ \mathbf{x}^{(2)[\bullet]_t} &\geq \mathbf{0}. \end{aligned}$$

Recall from equation [5.37] that $\mathbf{B}^{(1,0)[]}$ is a zero matrix so that the term $\mathbf{B}^{(1,0)[\bullet]_t} \mathbf{x}^{(0)[\bullet]_{t-1}}$ in the first set of detailed level constraints of problem [5.46] is absent in the first period, $t = 1$.

Dual problem **DMPGLP** [5.43] may also be equivalently formulated as:

$$z = \min \sum_{t=1}^{T} \sum_{h_t=1}^{H_t} \mathbf{b}'^{(0)[\bullet]_t} \boldsymbol{\pi}^{(0)[\bullet]_t} + \sum_{t=1}^{T} \sum_{h_t=1}^{H_t} \left(\mathbf{b}'^{(1)[\bullet]_t} \boldsymbol{\pi}^{(1)[\bullet]_t} + \mathbf{b}'^{(2)[\bullet]_t} \boldsymbol{\pi}^{(2)[\bullet]_t} \right) \tag{5.47}$$

s.t. aggregate level constraints at nodes $[\bullet]_t = (t, h_t)$, $h_t = 1, \ldots, H_t$, $t = 1, \ldots, T$,

$$\begin{aligned} \mathbf{W}'^{(0,0)[\bullet]_t}\boldsymbol{\pi}^{(0)[\bullet]_t} + \mathbf{W}'^{(1,0)[\bullet]_t}\boldsymbol{\pi}^{(1)[\bullet]_t} + \mathbf{W}'^{(2,0)[\bullet]_t}\boldsymbol{\pi}^{(2)[\bullet]_t} \\ + \sum_{l_t=1}^{L_t} \mathbf{B}'^{(1,0)[\bullet]_{t+1}}\boldsymbol{\pi}^{(1)[\bullet]_{t+1}} \geq \mathbf{c}^{(0)[\bullet]_t}, \\ \boldsymbol{\pi}^{(0)[\bullet]_t} \geq \mathbf{0}, \end{aligned}$$

and detailed level constraints at nodes $[\bullet]_t = (t, h_t)\,, h_t = 1, \ldots, H_t, t = 1, \ldots, T,$

$$\begin{aligned} \mathbf{W}'^{(1,1)[\bullet]_t}\boldsymbol{\pi}^{(1)[\bullet]_t} &\geq \mathbf{c}^{(1)[\bullet]_t}, \\ \mathbf{W}'^{(2,0)[\bullet]_t}\boldsymbol{\pi}^{(2)[\bullet]_t} &\geq \mathbf{c}^{(2)[\bullet]_t}, \\ \boldsymbol{\pi}^{(1)[\bullet]_t} &\geq \mathbf{0}, \\ \boldsymbol{\pi}^{(2)[\bullet]_t} &\geq \mathbf{0}. \end{aligned}$$

Recall from equation [5.37] that $\mathbf{B}^{(1,0)[\bullet]_{T+1}}$ are zero matrices so that the summation term in the first set of aggregate level constraints of problem [5.47] is absent in the terminal period, $t = T$.

Note that both the primal formulation [5.46] and the dual formulation [5.47] have the form of a two-stage stochastic linear program. Aggregate level constraints in both formulations are in the first stage while the detailed level constraints are in the second stage. Therefore, algorithm **LSD(multicut)** described in Section 3.1.4 can be applied directly to problem **PMPGLP** as formulated by [5.46]. Similarly, algorithm **DWD(multiactivities)** described in Section 3.2.3 can be applied directly to problem **DMPGLP** as formulated by [5.47]. Block-separable recourse allows for many other structural rearrangements that can be exploited for decomposition. Additional rearrangements are described in Section 5.4 below.

5.3.4 Sizes of Problems

The size of a model **MIMPSLP** primal or dual problem is expressed in terms of four parameters: number of variables, constraints, non-zero technology matrix coefficients, and nodes in the decision tree. Size parameters are derived in Appendix D for primal problem **PMPGLP** [5.19] and dual problem **DMPGLP** [5.24]. Table 5.3 summarizes the results obtained in Appendix D and is provided as a convenient reference.

5.4 Decomposition of Model Problems

Problems **PMPGLP** [5.19] (or [5.42] in array notation) and **DMPGLP** [5.24] ([5.43] in array notation) of model **MIMPSLP** are especially amenable to the decomposition procedures described in Chapters 3 and 4. An extensive selection of problem formulations and decomposition techniques are available due to the three model properties described in Section 5.3 above. The single-period problem is a stochastic linear program since an observation-terminated process is simulated by the model. Single-period problems, as well as problems with multiple periods, may be solved with a grand LP formulation or with DWD/LSD. Initialization of DWD/LSD algorithms is a simple process since model problems are bounded and have complete recourse. Complete recourse also insures that there will be no need for feasibility cuts in LSD or extreme direction activities in DWD. Furthermore, myopic decomposition may be an attractive alternative to DWD/LSD due to complete recourse. Block-separable recourse allows for significant flexibility in structuring

Table 5.3: Sizes of Model **MIMPSLP** Problems

Parameter	Symbol	Value	Equation
Nodes (Period)[a]	H_t	$\prod_{j=1}^{t-1} L_j$, where $H_1 = \prod_{j=1}^{0} L_j = 1$	[D.1]
Nodes (Cumulative)[b]	$H^{(t)}$	$\sum_{j=1}^{t} H_j = \sum_{j_1=1}^{t} \left(\prod_{j_2=1}^{j_1-1} L_{j_2} \right)$	[D.2]
Primal Constraints	$\#_{\mathsf{C}}^{\mathsf{P}}$	$(1+2N)\, H^{(T)} + \sum_{t=1}^{T} H_t L_t$	[D.3]
Primal Variables	$\#_{\mathsf{V}}^{\mathsf{P}}$	$(K+1)\, N H^{(T)} + Q \sum_{t=1}^{T} H_t L_t$	[D.6]
Primal Non-zeros[c]	$\#_{\mathsf{Z}}^{\mathsf{P}}$	$\left[(2K+5)\, H^{(T)} - 2\right] N + (N+Q) \sum_{t=1}^{T} H_t L_t$	[D.8]
Dual Constraints	$\#_{\mathsf{C}}^{\mathsf{D}}$	$(K+1)\, N H^{(T)} + Q \sum_{t=1}^{T} H_t L_t$	[D.7]
Dual Variables	$\#_{\mathsf{V}}^{\mathsf{D}}$	$[1 + (K+1)N]\, H^{(T)} + Q \sum_{t=1}^{T} H_t L_t$	[D.5]
Dual Non-zeros[d]	$\#_{\mathsf{Z}}^{\mathsf{D}}$	$\left[(3K+4)\, H^{(T)} - 2\right] N + (N+2Q-1) \sum_{t=1}^{T} H_t L_t$	[D.9]

[a] Number of nodes in period t, $1 \leq t \leq T$

[b] Number of cumulative nodes in periods 1 through t, $1 \leq t \leq T$

[c] Number of non-zeros in the primal technology matrix

[d] Number of non-zeros in the dual technology matrix

problems for DWD or LSD. Any model **MIMPSLP** problem may be treated as a two-stage problem regardless of the number of periods in the planning horizon. Problems involving three or more periods, on the other hand, may also be solved with nested decomposition modified to take advantage of block-separability. Each of these solution techniques is described in detail in Appendix E.

5.5 Implementation

Model **MIMPSLP** is implemented with a collection of nine computer code libraries. Linear programming support is provided by the commercial CPLEX[1] 6.5 callable library package. Random scenarios are generated using a Visual C++[2] 6.0 library, *MRScenGen*, developed and provided by a private company. Scenario generation is discussed in detail in the following section. The remaining seven libraries were developed or modified specifically for the **MIMPSLP** model. All routines in these latter seven libraries are written in either Visual C++ 6.0 or Compaq[3] Visual Fortran 6.1A (CVF). CVF is a Fortran 95 compiler with language extensions that allow fairly easy intercommunication between Fortran and C/C++ routines.

The seven model **MIMPSLP** specific libraries as well as the scenario generation library are listed in Table 5.4 with the library names and sizes for both the source and com-

---

1 CPLEX is a registered trademark of ILOG, Inc., Gentilly, France

2 Visual C++ is a registered trademark of Microsoft Corporation, Redmond, WA

3 Compaq is a registered trademark of Compaq Computer Corporation, Houston, TX

Table 5.4: Model **MIMPSLP** Run-Time Libraries

Description	**Library Name**	**Source Size (KB[a])**	**Compiled Size (KB)[b]**
Market Investment Model[c]	*MIMPSLP*	9,568.9	12,134.0
Options	*Options*	378.0	392.6
C++ Interface	*C++_Ifaces*	508.8	567.0
CPLEX Interface	*CPLEX_Ifaces*	326.2	355.2
Common	*Common*	328.9	552.3
Utilities	*Utils*	2,982.6	3,413.5
Decision Tree	*Stoc_Forest*	151.4	105.6
Scenario Generator[d]	*MRScenGen*	unknown	8,574.4
Combined Size		14,244.8[e]	26,094.6

[a] Kilobytes (1 kilobyte = $2^{10} = 1,024$ bytes)

[b] Each size includes both the static (*.lib*) and dynamic (*.dll*) libraries

[c] Includes the model routines as well as the routines' interfaces module and the version module

[d] Provided by a private company

[e] Without the scenario generator source files

piled files. Figure 5.5 is a schematic of the general calling sequence between main programs and the libraries. Note that library *Stoc_Forest* encapsulates the data and procedures required to implement the decision tree structure and is described in Section 4.6.

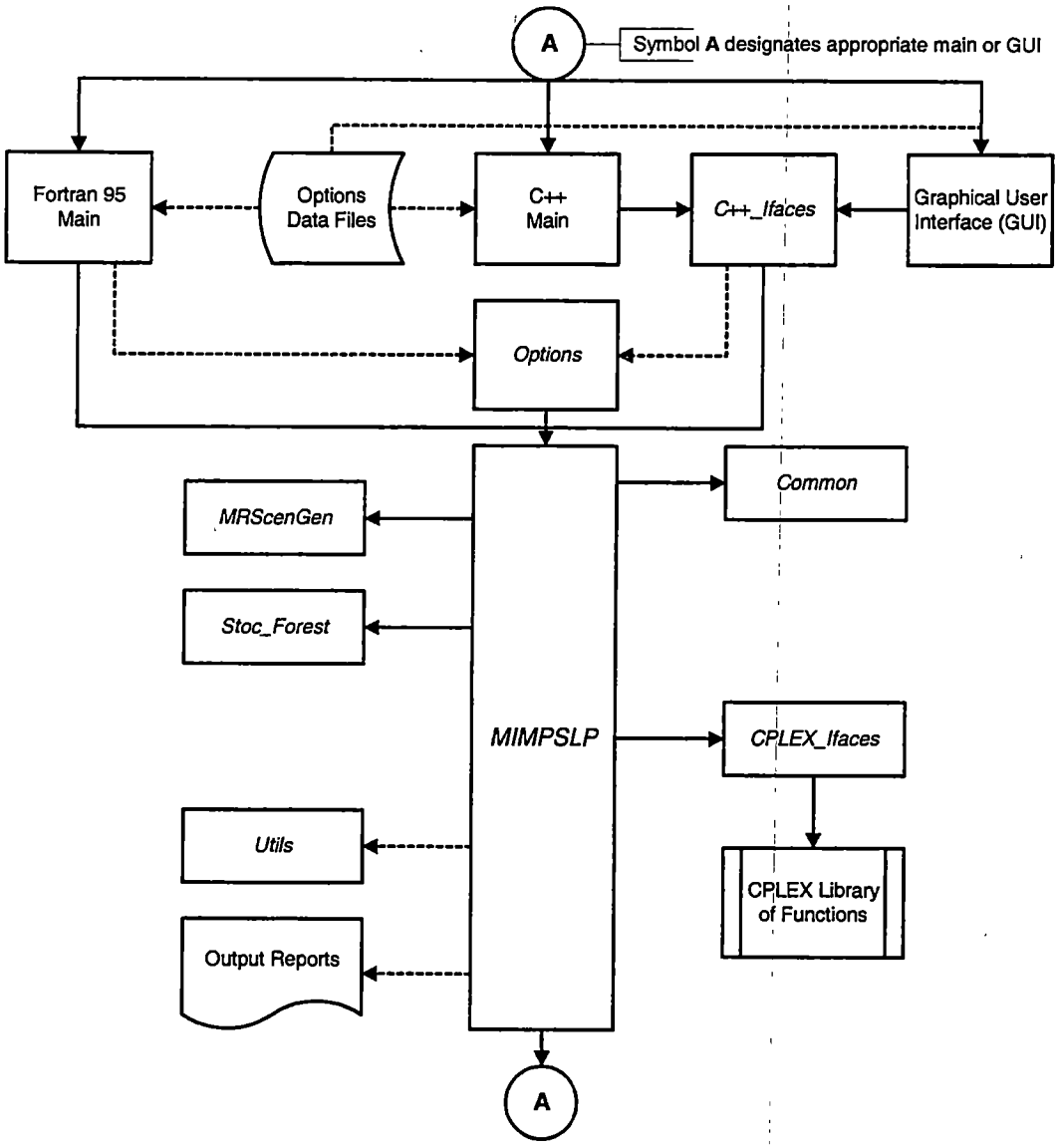
5.6 Scenario Generation

Scenario generation for model **MIMPSLP** is based upon the following assumptions.

Scenario Generation Assumptions

1. Daily returns for the N securities in each period form independent, identically distributed random vectors from a N-variate non-singular normal probability distribution.
2. The mean and standard deviation vectors and the correlation matrix for the daily returns in the first period of the planning horizon are known.
3. The distribution of daily returns in a given period is dependent on the observed returns in previous periods with a known dependence relationship.

Procedures, based upon the above assumptions, for generating random returns in a given period, creating conditional distributions, and drawing a random sample are described below. The procedural descriptions are followed by a discussion of the method used to measure sample effectiveness in approximating the derived distributions.



General Calling Sequence

1. Mains or GUI may leave default options in place or set all options with optional data files or set individual options with in-place code (C++ main or GUI must use *C++_Ifaces* library)
2. Mains or GUI invoke the *MIMPSLP* library (C++ main or GUI must use *C++_Ifaces* library) which retrieves options from the *Options* library and:
   a. Calls the *Common* library to set common data such as slippage piece-wise linear slopes and break points
   b. Creates the decision tree by node by alternately calling the *MRScenGen* library to generate the data and the *Stoc_forest* library to store the data as required by the number of periods and number of outcomes per period
   c. Solves problem with CPLEX using the intermediate *CPLEX_Ifaces* library
   d. Creates any optional output reports using applicable routines in the *Utils* library
   e. Destroys the decision tree and frees associated memory with call to the *Stoc_Forest* library
3. *MIMPSLP* returns to appropriate main or GUI

Figure 5.5: Schematic of Market Investment Model Libraries

5.6.1 Returns for a Period

The probability distribution for returns in a period is based upon the distribution of daily returns in that period. The following description for the derivation of the former distribution given the latter is for a generic period and subscripts/superscripts identifying the period are omitted for simplicity.

Let $\mathbf{y} \in \mathbb{R}^N$ be the vector of held positions and let $\hat{\mathbf{r}}_m \in \mathbb{R}^N$ represent the random vector of daily returns for day $m \in \{1, \ldots, \pounds\}$ in a period of length $\pounds \geq 1$ days. Total return over the period is then the random variable

$$TR = \hat{\mathbf{r}}_1'\mathbf{y} + \cdots + \hat{\mathbf{r}}_m'\mathbf{y} + \cdots + \hat{\mathbf{r}}_{\pounds}'\mathbf{y} = \left(\sum_{m=1}^{\pounds} \hat{\mathbf{r}}_m\right)' \mathbf{y}.$$

Returns for the period are therefore represented by the random vector

$$\mathbf{r} = \sum_{m=1}^{\pounds} \hat{\mathbf{r}}_m.$$

Let $\hat{\boldsymbol{\mu}} \in \mathbb{R}^N$, $\hat{\boldsymbol{\sigma}} \in \mathbb{R}^N$, and $\hat{\Gamma} \in \mathbb{R}^{N \times N}$ be the mean vector, standard deviation vector, and correlation matrix respectively for daily returns in the period. Scenario generation assumption number one implies that the random vector $\mathbf{r}$ is normally distributed with mean vector $\boldsymbol{\mu} \in \mathbb{R}^N$ and covariance matrix $\Sigma \in \mathbb{R}^{N \times N}$ such that (e.g., see Rencher [173, property 7, page 100]):

$$\begin{aligned} \boldsymbol{\mu} &= \sum_{m=1}^{\pounds} \hat{\boldsymbol{\mu}} = \pounds \hat{\boldsymbol{\mu}}, \\ \Sigma &= \sum_{m=1}^{\pounds} \hat{\Sigma} = \pounds \hat{\Sigma}, \end{aligned}$$

where $\hat{\Sigma}$ is the covariance matrix for daily returns. Covariance matrices must be determined using the corresponding correlation matrices and the well known relationship (e.g., see Rencher [173, equation 3.36, page 69])

$$\hat{\Sigma} = \mathbf{D}_{\hat{\sigma}} \hat{\Gamma} \mathbf{D}_{\hat{\sigma}},$$

where $\mathbf{D}_{\mathbf{v}} \in \mathbb{R}^{M \times M}$ represents the diagonal matrix formed with vector $\mathbf{v} \in \mathbb{R}^{M}$:

$$(D_{\mathbf{v}})_{ij} = \begin{cases} v_i, & i = j, \\ 0, & i \neq j. \end{cases}$$

Note that matrix-vector multiplication is not required for the product $\mathbf{D}_{\mathbf{v}}\mathbf{u}$ where $\mathbf{u} \in \mathbb{R}^{M}$ is compatible with $\mathbf{D}_{\mathbf{v}}$. The same result may be achieved with the *Hadamard* product denoted by the symbol $\odot$:

$$\mathbf{D}_{\mathbf{v}}\mathbf{u} = \mathbf{v} \odot \mathbf{u} = (v_1 u_1, \ldots, v_m u_m, \ldots, v_M u_M)'. \qquad [5.48]$$

Several properties of the Hadamard product are described by Magnus and Neudecker [139, Chapter 3, Section 6]. This product will be used in the description of conditional distributions in Section 5.6.2 below.

The covariance matrix for period returns is then

$$\Sigma = \pounds \mathbf{D}_{\hat{\sigma}} \hat{\Gamma} \mathbf{D}_{\hat{\sigma}} = \left(\sqrt{\pounds}\mathbf{D}_{\hat{\sigma}}\right) \hat{\Gamma} \left(\sqrt{\pounds}\mathbf{D}_{\hat{\sigma}}\right) = \mathbf{D}_{\sigma} \hat{\Gamma} \mathbf{D}_{\sigma},$$

where $\sigma = \sqrt{\pounds}\hat{\sigma}$ is the vector of standard deviations for the returns in the period. Note that the above relationship also demonstrates that the correlation matrix for period returns is the same as that for the daily returns. Therefore, random vector $\mathbf{r}$, representing the returns for the period, has the N-variate normal distribution $\mathcal{N}_N\left(\mu, \mathbf{D}_{\sigma} \hat{\Gamma} \mathbf{D}_{\sigma}\right)$.

5.6.2 Conditional Distributions

The distribution of returns for the first period is determined with the procedure described above using the parameters for the first period daily returns distribution. First period daily returns distribution parameters are known by the second scenario generation assumption. Daily returns distributions for periods after the first are dependent on the observed returns in previous periods. Procedures for creating the conditional distributions given the dependence relationship introduced by the third scenario generation assumption are discussed below.

Let $\hat{\boldsymbol{\mu}}^{[]} \in \mathbb{R}^N$, $\hat{\boldsymbol{\sigma}}^{[]} \in \mathbb{R}^N$, and $\hat{\boldsymbol{\Gamma}}^{[]} \in \mathbb{R}^{N \times N}$ be the known mean vector, standard deviation vector, and positive definite correlation matrix respectively for daily returns in the first period. Assume that the columns of history matrix

$$\mathbf{R}^{\mathrm{H}}_{\bar{t}-1} = [\mathbf{r}_1, \ldots, \mathbf{r}_{\bar{t}-1}] \in \mathbb{R}^{N \times (\bar{t}-1)}$$

are the observed returns in periods 1 through $\bar{t}-1$ where $2 \leq \bar{t} \leq T$ and period t has length $\mathcal{L}_t \geq 1$ days for $t = 1, \ldots, \bar{t}$. The conditional distribution for returns at some node $[\bullet]_{\bar{t}}$ in period $\bar{t}$ given history matrix $\mathbf{R}^{\mathrm{H}}_{\bar{t}-1}$ must now be determined. Conditional distributions in model **MIMPSLP** are based upon a dependence relationship that:

1. sets the standard deviation vector and correlation matrix for daily returns at the period $\bar{t}$ node equal to the corresponding arrays for the first period node:

$$\hat{\boldsymbol{\sigma}}^{[\bullet]_{\bar{t}}} = \hat{\boldsymbol{\sigma}}^{[]} \text{ and } \hat{\boldsymbol{\Gamma}}^{[\bullet]_{\bar{t}}} = \hat{\boldsymbol{\Gamma}}^{[]}, \text{ and} \qquad [5.49]$$

2. sets the mean vector for daily returns at the period $\bar{t}$ node according to:

$$\hat{\boldsymbol{\mu}}^{[\bullet]_{\bar{t}}} = \hat{\boldsymbol{\mu}}^{[]} + f\left(\mathbf{v} - \hat{\boldsymbol{\mu}}^{[]}\right), \qquad [5.50]$$

where the dependence factor f and the historical conditioning vector $\mathbf{v}$ are described below.

Dependence factor f in equation [5.50] is a constant, $0 \leq f \leq 1$, that determines the degree of dependency on the history of the process. Daily returns distributions are independent of the history when $f = 0$ since the parameters for the distributions in each period remain constant at their first period values. Conversely, the mean vector for the daily returns distribution at a node in period $\bar{t}$, $2 \leq \bar{t} \leq T$, is equal to the historical conditioning vector $\mathbf{v}$ (following description) when $f = 1$. Otherwise, $0 < f < 1$ and the daily returns mean vector at node $[\bullet]_{\bar{t}}$ is a function of the historical conditioning vector and the first period daily returns mean vector.

Historical conditioning vector $\mathbf{v}$ is a weighted sum of the columns the history matrix $\mathbf{R}^{H}_{\bar{t}-1}$ scaled by the corresponding period lengths:

$$\mathbf{v} = \sum_{t=1}^{\bar{t}-1} w_t \left(\frac{\mathbf{r}_t}{\mathcal{L}_t}\right), \qquad [5.51]$$

where,

$$\sum_{t=1}^{\bar{t}-1} w_t = 1. \qquad [5.52]$$

Returns in period t are considered to be twice as influential as the returns in period $t-1$ so that the additional conditions

$$w_t = 2w_{t-1}, t = 2, \ldots, \bar{t} - 1 \qquad [5.53]$$

are imposed on the weights. Equation [5.53] implies that the weights form a geometric progression with (e.g., see the CRC Math Handbook [39, equation 2.1.4]):

$$\sum_{t=1}^{\bar{t}-1} w_t = \sum_{t=1}^{\bar{t}-1} w_{\bar{t}-t} = w_{\bar{t}-1}\frac{1-\left(\frac{1}{2}\right)^{\bar{t}-1}}{\frac{1}{2}}, \qquad [5.54]$$

where the second term above simply reverses the order of the weights in the first term. Equations [5.52] through [5.54] then yield the weights,

$$w_t = \begin{cases} \frac{1}{2\left[1-\left(\frac{1}{2}\right)^{\bar{t}-1}\right]}, & t = \bar{t}-1 \\ \\ \frac{1}{2}w_{t+1}, & t = \bar{t}-2, \ldots 1, \end{cases}$$

used in equation [5.51] to determine the historical conditioning vector.

Equations [5.49] and [5.50] can now be used to determine the daily returns parameters $\hat{\boldsymbol{\sigma}}^{[\bullet]_{\bar{t}}}$ and $\hat{\boldsymbol{\mu}}^{[\bullet]_{\bar{t}}}$ respectively. Random vector $\mathbf{r}^{[\bullet]_{\bar{t}}}$, representing the returns for a node in period $\bar{t}$, $2 \leq \bar{t} \leq T$, then has the N-variate normal distribution $\mathcal{N}_N\left(\boldsymbol{\mu}^{[\bullet]_{\bar{t}}}, \mathbf{D}_{\boldsymbol{\sigma}^{[\bullet]_{\bar{t}}}}\hat{\boldsymbol{\Gamma}}^{[]}\mathbf{D}_{\boldsymbol{\sigma}^{[\bullet]_{\bar{t}}}}\right)$ where $\boldsymbol{\mu}^{[\bullet]_{\bar{t}}} = \mathcal{L}_{\bar{t}}\hat{\boldsymbol{\mu}}^{[\bullet]_{\bar{t}}}$ and $\boldsymbol{\sigma}^{[\bullet]_{\bar{t}}} = \sqrt{\mathcal{L}_{\bar{t}}}\hat{\boldsymbol{\sigma}}^{[\bullet]_{\bar{t}}}$.

5.6.3 Sampling

Downside deviation at each node in the decision tree is approximated by sampling from the N-variate normal distribution of returns at the node. Assume that the N-variate normal distribution $\mathcal{N}_N\left(\boldsymbol{\mu}^{[\bullet]_t}, \mathbf{D}_{\boldsymbol{\sigma}^{[\bullet]_t}}\hat{\boldsymbol{\Gamma}}^{[]}\mathbf{D}_{\boldsymbol{\sigma}^{[\bullet]_t}}\right)$ for the returns at a node in period t, $1 \leq t \leq T$, have been determined as described in Section 5.6.2 above. Covariance matrix $\boldsymbol{\Sigma}^{[\bullet]_t} = \mathbf{D}_{\boldsymbol{\sigma}^{[\bullet]_t}}\hat{\boldsymbol{\Gamma}}^{[]}\mathbf{D}_{\boldsymbol{\sigma}^{[\bullet]_t}}$ is symmetric and positive definite due to the non-singular normality assumption. Therefore, there exists a lower triangular Cholesky decomposition matrix $\mathbf{C}^{[\bullet]_t} \in \mathbb{R}^{N\times N}$ such that $\mathbf{C}^{[\bullet]_t}\mathbf{C}'^{[\bullet]_t} = \boldsymbol{\Sigma}^{[\bullet]_t}$ (e.g., see Golub and Van Loan [93, Section

4.2.3] or Rencher [173, Section 2.7]). Then, given a vector $\mathbf{u} \in \mathbb{R}^N$ of N independent samples from a standard univariate normal distribution, DeGroot [52, Section 5.4], Krzanowski [130, Section 7.2] and Rencher [173, Section 4.2] show that

$$\mathbf{r} = \mathbf{C}^{[\bullet]_t}\mathbf{u} + \boldsymbol{\mu}^{[\bullet]_t} \qquad [5.55]$$

is a sample vector from $\mathcal{N}_N\left(\boldsymbol{\mu}^{[\bullet]_t}, \mathbf{D}_{\boldsymbol{\sigma}^{[\bullet]_t}}\hat{\boldsymbol{\Gamma}}^{[]}\mathbf{D}_{\boldsymbol{\sigma}^{[\bullet]_t}}\right)$. Since a constant correlation matrix $\hat{\boldsymbol{\Gamma}}^{[]}$ (see the first dependence condition on page 211) across all periods, equation [5.55] can be replaced by

$$\mathbf{r} = \mathbf{D}_{\boldsymbol{\sigma}^{[\bullet]_t}}\left(\hat{\mathbf{C}}\mathbf{u}\right) + \boldsymbol{\mu}^{[\bullet]_t}, \qquad [5.56]$$

where $\hat{\mathbf{C}}$ is the lower triangular Choleksy decomposition matrix for $\hat{\boldsymbol{\Gamma}}^{[]}$. Equation [5.56] is equivalent to equation [5.55] since,

$$\begin{aligned}
&\boldsymbol{\Sigma}^{[\bullet]_t} = \mathbf{D}_{\boldsymbol{\sigma}^{[\bullet]_t}}\hat{\boldsymbol{\Gamma}}^{[]}\mathbf{D}_{\boldsymbol{\sigma}^{[\bullet]_t}} \\
\Longrightarrow\ &\mathbf{C}^{[\bullet]_t}\mathbf{C}'^{[\bullet]_t} = \mathbf{D}_{\boldsymbol{\sigma}^{[\bullet]_t}}\hat{\mathbf{C}}\hat{\mathbf{C}}'\mathbf{D}_{\boldsymbol{\sigma}^{[\bullet]_t}} \\
\Longrightarrow\ &\mathbf{C}^{[\bullet]_t}\mathbf{C}'^{[\bullet]_t} = \left(\mathbf{D}_{\boldsymbol{\sigma}^{[\bullet]_t}}\hat{\mathbf{C}}\right)\left(\mathbf{D}_{\boldsymbol{\sigma}^{[\bullet]_t}}\hat{\mathbf{C}}\right)' \\
\Longrightarrow\ &\mathbf{C}^{[\bullet]_t} = \mathbf{D}_{\boldsymbol{\sigma}^{[\bullet]_t}}\hat{\mathbf{C}}.
\end{aligned}$$

Use of equation [5.56] in lieu of equation [5.55] saves considerable time since only one Cholesky decomposition matrix must be determined and operation $\mathbf{D}_{\boldsymbol{\sigma}^{[\bullet]_t}}\left(\hat{\mathbf{C}}\mathbf{u}\right) = \boldsymbol{\sigma}^{[\bullet]_t} \odot \left(\hat{\mathbf{C}}\mathbf{u}\right)$ is significantly faster than the Cholesky decomposition procedure.

Assume the next node to add to the decision tree is in period t, $1 \leq t \leq T$, with length $\pounds_t \geq 1$ days. Let the path vector to the node be $[\bullet]_t = \left[\tilde{l}_1, \ldots, \tilde{l}_{t-1}\right]$ which implies

the returns history matrix $\mathbf{R}_{t-1}^{\mathrm{H}} = \left[\mathbf{R}_{\bullet \tilde{l}_1}^{[]}, \ldots, \mathbf{R}_{\bullet \tilde{l}_{t-1}}^{[\bullet]_{t-1}}\right]$ where $\mathbf{R}_0^{\mathrm{H}} = [\,]$ (i.e., empty history matrix if $t = 1$). Then, the following procedure could be used to draw a sample of L_t returns.

Cholesky Sampling Procedure

Step 1: Derive the unconditional ($t = 1$) or conditional ($2 \leq t \leq T$) distribution $\mathcal{N}_N\left(\boldsymbol{\mu}^{[\bullet]_t}, \mathbf{D}_{\boldsymbol{\sigma}^{[\bullet]_t}} \hat{\boldsymbol{\Gamma}}^{[]} \mathbf{D}_{\boldsymbol{\sigma}^{[\bullet]_t}}\right)$ using the procedures described in Sections 5.6.1 and 5.6.2. Set $l \leftarrow 0$ and go to Step 2.

Step 2: If $l = L_t$, return to invoking procedure. Set $l \leftarrow l + 1$ and create a vector $\mathbf{u} \in \mathbb{R}^{N \times L_t}$ of uniform random numbers from $(0, 1)$. Transform $\mathbf{u}$ into a vector of random numbers from the standard normal distribution using the inverse distribution procedure and go to Step 3.

Step 3: Determine the next column for $\mathbf{R}^{[\bullet]_t}$:

$$\mathbf{R}_{\bullet l}^{[\bullet]_t} = \boldsymbol{\sigma}^{[\bullet]_t} \odot \left(\hat{\mathbf{C}}\mathbf{u}\right) + \boldsymbol{\mu}^{[\bullet]_t}.$$

Return to Step 2.

Methods other than the inverse distribution procedure may be used in Step 2 to obtain a vector of standard normal values. The National Bureau of Standards Math Handbook [156, Section 26.8, pages 952-953] summarizes the inverse distribution, sum of uniform deviates, direct, and acceptance-rejection methods for drawing a standard normal sample.

5.6.4 Sample Effectiveness

Mahalanobis squared distances (MSD) (see Krzanowski [130, page 234] and Rencher [173, Section 3.12]) are used to measure the effectiveness of a sample in approximating the parent distribution. The MSD between two vectors, $\mathbf{v} \in \mathbb{R}^M$ and $\mathbf{w} \in \mathbb{R}^M$, associated with a distribution with positive definite covariance matrix $\mathbf{\Sigma}_M \in \mathbb{R}^{M \times M}$ is denoted by $\mathcal{M}_{\mathbf{\Sigma}_M}(\mathbf{v}, \mathbf{w})$ and is defined to be:

$$\mathcal{M}_{\mathbf{\Sigma}_M}(\mathbf{v}, \mathbf{w}) = (\mathbf{v} - \mathbf{w})' \, \mathbf{\Sigma}_M^{-1} (\mathbf{v} - \mathbf{w}). \qquad [5.57]$$

The MSD for one dimensional components is simply the square of the ratio of the distance between two points, v and w, and the standard deviation, σ, of the distribution:

$$\mathcal{M}_{\mathbf{\Sigma}_1}(v, w) = \left(\frac{|v - w|}{\sigma} \right)^2,$$

where $\mathbf{\Sigma}_1 = [\sigma^2]$.

Mahalanobis squared distances, or simply Mahalanobis distances, are used frequently in statistical data analysis. Bonanno and Griffiths [24] use a multivariate discrimination technique based upon Mahalanobis distances to discriminate between alcohols. Dai and Khorram [54] use Mahalanobis distances to detect image misregistration in order to improve the accuracy of multisource data analysis. Kato et al. [119] develop a system for recognizing handwritten Chinese and Japanese characters based upon modified Mahalanobis distances. Lui and Cheng [136] maximize the Mahalanobis distance as part of procedure to select the number of subimages in an image segmentation problem. Penny [163] discusses the use of Mahalanobis distances in detecting outliers in multivariate data.

Chapter 5 Market Investment Model

Sample effectiveness in approximating the parent distribution is measured herein by comparing the population expected values of selected Mahalanobis squared distances with the corresponding average values realized from the sample. Three Mahalanobis squared distances for each of two model random quantities, returns and net returns, are used in the analysis:

1. MSD between the random quantity and the population mean of that quantity,
2. MSD between the sample average of the random quantity with the population mean of that quantity, and
3. MSD between the random quantity and the sample average of the random quantity.

Evaluating the mean of a MSD is a necessary procedure in the analysis. Let $\boldsymbol{\mu} \in \mathbb{R}^M$ and $\boldsymbol{\Sigma}_M \in \mathbb{R}^{M \times M}$ be the mean vector and positive definite covariance matrix respectively for some M-variate distribution. Then, if $\mathbf{v} \in \mathbb{R}^M$ represents a random vector from the distribution, the expected value of the first MSD to be determined is:

$$\mathcal{E}\left[\mathcal{M}_{\boldsymbol{\Sigma}_M}(\mathbf{v}, \boldsymbol{\mu})\right] = \mathcal{E}\left[(\mathbf{v} - \boldsymbol{\mu})' \boldsymbol{\Sigma}_M^{-1} (\mathbf{v} - \boldsymbol{\mu})\right].$$

Krzanowski [130, Section 7.5, page 212] and Rencher [173, Section 4.2, property 3] show that $\mathcal{M}_{\boldsymbol{\Sigma}_M}(\mathbf{v}, \boldsymbol{\mu})$ has a Chi-square distribution with M degrees of freedom when the distribution is M-variate normal. Therefore, when the parent distribution is M-variate normal, the mean and variance, denoted by $\mathcal{V}(\bullet)$, of $\mathcal{M}_{\boldsymbol{\Sigma}_M}(\mathbf{v}, \boldsymbol{\mu})$ are (e.g., see Lindgren [132, Section 6.9]):

$$\mathcal{E}\left[\mathcal{M}_{\boldsymbol{\Sigma}_M}(\mathbf{v}, \boldsymbol{\mu})\right] = M,$$

and

$$\mathcal{V}\left[\mathcal{M}_{\boldsymbol{\Sigma}_M}(\mathbf{v}, \boldsymbol{\mu})\right] = 2M.$$

The latter two expressions above could be used in the analysis since library *MRScen-Gen* assumes a N-variate non-singular normal distribution. Model **MIMPSLP** does not, however, rely upon this condition so that expressions for a more general distribution are desired. Evaluations of the three Mahalanobis squared distances described above use the following three propositions.

Proposition 13 *Let $\mathbf{v} \in \mathbb{R}^M$ be a random vector from a M-variate distribution with mean vector $\boldsymbol{\mu} \in \mathbb{R}^M$ and positive definite covariance matrix $\boldsymbol{\Sigma}_M \in \mathbb{R}^{M \times M}$. Then,*

$$\mathcal{E}\left[\mathcal{M}_{\boldsymbol{\Sigma}_M}(\mathbf{v}, \boldsymbol{\mu})\right] = M. \qquad [5.58]$$

Proof See Appendix F.

Proposition 14 *Let $\bar{\mathbf{v}} \in \mathbb{R}^M$ be the average random vector for a random sample of size L drawn from a M-variate distribution with mean vector $\boldsymbol{\mu} \in \mathbb{R}^M$ and positive definite covariance matrix $\boldsymbol{\Sigma}_M \in \mathbb{R}^{M \times M}$. Then,*

$$\mathcal{E}\left[\mathcal{M}_{\boldsymbol{\Sigma}_M}(\bar{\mathbf{v}}, \boldsymbol{\mu})\right] = \frac{M}{L}, \qquad [5.59]$$

when sampling is performed with replacement if the distribution population is discrete.

Proof See Appendix F.

Proposition 15 *Let $\bar{\mathbf{v}} \in \mathbb{R}^M$ be the average random vector for a random sample of size L drawn from a M-variate distribution with mean vector $\boldsymbol{\mu} \in \mathbb{R}^M$ and positive definite covariance matrix $\boldsymbol{\Sigma}_M \in \mathbb{R}^{M \times M}$. If $\mathbf{v}$ is a representative vector from the random sample, then*

$$\mathcal{E}\left[\mathcal{M}_{\boldsymbol{\Sigma}_M}(\mathbf{v}, \bar{\mathbf{v}})\right] = \frac{L-1}{L} M, \qquad [5.60]$$

when sampling is performed with replacement if the distribution population is discrete.

Proof See Appendix F.

The expected Mahalanobis squared distances for random returns and random net returns are described below followed by proposed measures for sample effectiveness.

Expected MSD for Random Returns

The expected Mahalanobis squared distances for random returns at a node $[\bullet]_t$, $1 \leq t \leq T$, in the decision tree are determined with direct applications of equations [5.58] through [5.60]. Let $\mathbf{r}^{[\bullet]_t}$ represent a random vector in the sample $\mathbf{R}^{[\bullet]_t}$ (i.e., $\mathbf{r}^{[\bullet]_t}$ represents

a column from $\mathbf{R}^{[\bullet]_t} \in \mathbb{R}^{N \times L_t}$), $\bar{\mathbf{r}}$ be the average of the sample random vectors in $\mathbf{R}^{[\bullet]_t}$, and

$$\boldsymbol{\Sigma}_N^{[\bullet]_t} = \mathbf{D}_{\boldsymbol{\sigma}^{[\bullet]_t}} \hat{\boldsymbol{\Gamma}}^{[]} \mathbf{D}_{\boldsymbol{\sigma}^{[\bullet]_t}}.$$

The expected Mahalanobis squared distances at node $[\bullet]_t$, $1 \leq t \leq T$, are then:

$$\mathcal{E}\left[\mathcal{M}_{\boldsymbol{\Sigma}_N^{[\bullet]_t}}\left(\mathbf{r}^{[\bullet]_t}, \boldsymbol{\mu}^{[\bullet]_t}\right)\right] = N, \qquad [5.61]$$

$$\mathcal{E}\left[\mathcal{M}_{\boldsymbol{\Sigma}_N^{[\bullet]_t}}\left(\bar{\mathbf{r}}^{[\bullet]_t}, \boldsymbol{\mu}^{[\bullet]_t}\right)\right] = \frac{N}{L_t}, \qquad [5.62]$$

$$\mathcal{E}\left[\mathcal{M}_{\boldsymbol{\Sigma}_N^{[\bullet]_t}}\left(\mathbf{r}^{[\bullet]_t}, \bar{\mathbf{r}}^{[\bullet]_t}\right)\right] = \frac{L_t - 1}{L_t} N. \qquad [5.63]$$

Expected Mahalanobis squared distances for all nodes are evaluated directly with equations [5.61] through [5.63]. The expected squared distances for each period, denoted by $\mathcal{E}\left(\mathcal{M}_j^{(t)}\right)$, $j = 1, 2, 3$, for $t = 1, \ldots, T$, are then defined as:

$$\mathcal{E}\left(\mathcal{M}_1^{(t)}\right) = \sum_{h_t}^{H_t} \hat{p}^{[\bullet]_t} \mathcal{E}\left[\mathcal{M}_{\boldsymbol{\Sigma}_N^{[\bullet]_t}}\left(\mathbf{r}^{[\bullet]_t}, \boldsymbol{\mu}^{[\bullet]_t}\right)\right] = N \sum_{h_t}^{H_t} \hat{p}^{[\bullet]_t} = N, t = 1, \ldots, T, \qquad [5.64]$$

$$\mathcal{E}\left(\mathcal{M}_2^{(t)}\right) = \sum_{h_t}^{H_t} \hat{p}^{[\bullet]_t} \mathcal{E}\left[\mathcal{M}_{\boldsymbol{\Sigma}_N^{[\bullet]_t}}\left(\bar{\mathbf{r}}^{[\bullet]_t}, \boldsymbol{\mu}^{[\bullet]_t}\right)\right] = \frac{N}{L_t} \sum_{h_t}^{H_t} \hat{p}^{[\bullet]_t} = \frac{N}{L_t}, t = 1, \ldots, T, \qquad [5.65]$$

$$\mathcal{E}\left(\mathcal{M}_3^{(t)}\right) = \sum_{h_t}^{H_t} \hat{p}^{[\bullet]_t} \mathcal{E}\left[\mathcal{M}_{\boldsymbol{\Sigma}_N^{[\bullet]_t}}\left(\mathbf{r}^{[\bullet]_t}, \bar{\mathbf{r}}^{[\bullet]_t}\right)\right] = \frac{L_t - 1}{L_t} N \sum_{h_t}^{H_t} \hat{p}^{[\bullet]_t} = \frac{L_t - 1}{L_t} N, t = 1, \ldots, T, \qquad [5.66]$$

where equation [2.7] on page 28 is used to determine label $[\bullet]_t$ given label (t, h_t) throughout this discussion.

Expected values determined with equations [5.64] through [5.66] are compared with the corresponding average values realized from the samples to measure the effectiveness of the samples in approximating the parent distributions. Average values, denoted by $\mathcal{A}\left(\mathcal{M}_j^{(t)}\right)$, $j = 1, 2, 3$, for $t = 1, \ldots, T$, are determined using equation [5.57] and the drawn samples:

$$\mathcal{A}\left(\mathcal{M}_1^{(t)}\right) = \sum_{h_t}^{H_t} \hat{p}^{[\bullet]_t} \left[\sum_{l_t=1}^{L_t} p_{l_t}^{[\bullet]_t} \left(\mathbf{R}_{\bullet l_t}^{[\bullet]_t} - \boldsymbol{\mu}^{[\bullet]_t} \right)' \left(\boldsymbol{\Sigma}_N^{[\bullet]_t} \right)^{-1} \left(\mathbf{R}_{\bullet l_t}^{[\bullet]_t} - \boldsymbol{\mu}^{[\bullet]_t} \right) \right], t = 1, \ldots, T, \quad [5.67]$$

$$\mathcal{A}\left(\mathcal{M}_2^{(t)}\right) = \sum_{h_t}^{H_t} \hat{p}^{[\bullet]_t} \left[\sum_{l_t=1}^{L_t} p_{l_t}^{[\bullet]_t} \left(\bar{\mathbf{r}}^{[\bullet]_t} - \boldsymbol{\mu}^{[\bullet]_t} \right)' \left(\boldsymbol{\Sigma}_N^{[\bullet]_t} \right)^{-1} \left(\bar{\mathbf{r}}^{[\bullet]_t} - \boldsymbol{\mu}^{[\bullet]_t} \right) \right], t = 1, \ldots, T, \quad [5.68]$$

$$\mathcal{A}\left(\mathcal{M}_3^{(t)}\right) = \sum_{h_t}^{H_t} \hat{p}^{[\bullet]_t} \left[\sum_{l_t=1}^{L_t} p_{l_t}^{[\bullet]_t} \left(\mathbf{R}_{\bullet l_t}^{[\bullet]_t} - \bar{\mathbf{r}}^{[\bullet]_t} \right)' \left(\boldsymbol{\Sigma}_N^{[\bullet]_t} \right)^{-1} \left(\mathbf{R}_{\bullet l_t}^{[\bullet]_t} - \bar{\mathbf{r}}^{[\bullet]_t} \right) \right], t = 1, \ldots, T. \quad [5.69]$$

Expected MSD for Random Net Returns

Expected Mahalanobis squared distances for random net returns at a node $[\bullet]_t$, $1 \leq t \leq T$, in the decision tree are determined similarly to those for random returns. The primary difference in the two procedures is that the distributions for random net returns may not be N-dimensional. Net return for security $n \in \mathrm{N}$ is not a random variable if the number of held positions, $y_n^{[\bullet]_t}$, is zero since $y_n^{[\bullet]_t} R_{nl_t}^{[\bullet]_t} = 0$ for $l_t = 1, \ldots, L_t$. Propositions

13 through 15 can still be used to determine the expected squared distances for random net return as demonstrated below.

Define $\hat{\mathsf{N}}^{[\bullet]_t}$ as the index sets

$$\hat{\mathsf{N}}^{[\bullet]_t} = \left\{ n \,|\, y_n^{[\bullet]_t} \neq 0 \right\}, h_t = 1, \ldots, H_t, t = 1, \ldots, T,$$

and let $\hat{N}^{[\bullet]_t}$ be the number of indices in set $\hat{\mathsf{N}}^{[\bullet]_t}$, $t \in \{1, \ldots, T\}$. Let $\mathbf{y}^{[\bullet]_t}\left(\hat{\mathsf{N}}^{[\bullet]_t}\right) \in \mathbb{R}^{\hat{N}^{[\bullet]_t}}$ and $\boldsymbol{\mu}^{[\bullet]_t}\left(\hat{\mathsf{N}}^{[\bullet]_t}\right) \in \mathbb{R}^{\hat{N}^{[\bullet]_t}}$ denote the portfolio vector and mean returns vector respectively that include only securities with non-zero held positions at node $[\bullet]_t$, $1 \leq t \leq T$. Similarly, let $\mathbf{R}^{[\bullet]_t}\left(\hat{\mathsf{N}}^{[\bullet]_t}\right) \in \mathbb{R}^{\hat{N}^{[\bullet]_t} \times L_t}$ be the matrix of sample returns that includes only securities with non-zero held positions at each node. Define vectors $\hat{\boldsymbol{\mu}}_{\text{TR}}^{[\bullet]_t} \in \mathbb{R}^{\hat{N}^{[\bullet]_t}}$ and $\mathbf{u}_{l_t}^{[\bullet]_t} \in \mathbb{R}^{\hat{N}^{[\bullet]_t}}$ as:

$$\hat{\boldsymbol{\mu}}_{\text{TR}}^{[\bullet]_t} = \mathbf{y}^{[\bullet]_t}\left(\hat{\mathsf{N}}^{[\bullet]_t}\right) \odot \boldsymbol{\mu}^{[\bullet]_t}\left(\hat{\mathsf{N}}^{[\bullet]_t}\right), h_t = 1, \ldots, H_t, t = 1, \ldots, T,$$

and

$$\mathbf{u}_{l_t}^{[\bullet]_t} = \mathbf{y}^{[\bullet]_t}\left(\hat{\mathsf{N}}^{[\bullet]_t}\right) \odot \mathbf{R}_{\bullet l_t}^{[\bullet]_t}\left(\hat{\mathsf{N}}^{[\bullet]_t}\right), l_t = 1, \ldots L_t, h_t = 1, \ldots, H_t, t = 1, \ldots, T,$$

and let $\mathbf{u}^{[\bullet]_t}$ and $\bar{\mathbf{u}}^{[\bullet]_t}$ be a representative vector and the sample average vector respectively at each node. Finally, let $\hat{\boldsymbol{\Sigma}}_{\hat{N}^{[\bullet]_t}}^{[\bullet]_t} \in \mathbb{R}^{\hat{N}^{[\bullet]_t} \times \hat{N}^{[\bullet]_t}}$ be the covariance matrix at node $[\bullet]_t$, $1 \leq t \leq T$, obtained by deleting all rows and columns in $\boldsymbol{\Sigma}_N^{[\bullet]_t}$ with indices n such that $n \notin \hat{\mathsf{N}}^{[\bullet]_t}$.

Magnus and Neudecker [139, Theorem 29] indicate that covariance matrices $\hat{\boldsymbol{\Sigma}}_{\hat{N}^{[\bullet]_t}}^{[\bullet]_t}$ are positive definite given that the corresponding parent matrices $\boldsymbol{\Sigma}_N^{[\bullet]_t}$ are positive definite at each node. Furthermore, Rencher [173, Section 4.2, property 4a] states that $\mathbf{u}^{[\bullet]_t}$ will have a $\hat{N}^{[\bullet]_t}$-variate distribution with mean vector $\hat{\boldsymbol{\mu}}^{[\bullet]_t}$ and covariance matrix $\hat{\boldsymbol{\Sigma}}_{\hat{N}^{[\bullet]_t}}^{[\bullet]_t}$.

Propositions 13 through 15 then imply that the expected Mahalanobis squared distances for net returns, denoted by $\mathcal{E}\left(\mathcal{M}_j^{(t)}\right)$, $j = 4, 5, 6$, for periods $t = 1, \ldots, T$, are:

$$\mathcal{E}\left(\mathcal{M}_4^{(t)}\right) = \sum_{h_t}^{H_t} \hat{p}^{[\bullet]_t} \mathcal{E}\left[\mathcal{M}_{\hat{\boldsymbol{\Sigma}}_{\hat{N}^{[\bullet]_t}}^{[\bullet]_t}}\left(\mathbf{u}^{[\bullet]_t}, \hat{\boldsymbol{\mu}}_{\mathrm{TR}}^{[\bullet]_t}\right)\right] = \sum_{h_t}^{H_t} \hat{p}^{[\bullet]_t} \hat{N}^{[\bullet]_t}, t = 1, \ldots, T, \quad [5.70]$$

$$\mathcal{E}\left(\mathcal{M}_5^{(t)}\right) = \sum_{h_t}^{H_t} \hat{p}^{[\bullet]_t} \mathcal{E}\left[\mathcal{M}_{\hat{\boldsymbol{\Sigma}}_{\hat{N}^{[\bullet]_t}}^{[\bullet]_t}}\left(\bar{\mathbf{u}}^{[\bullet]_t}, \hat{\boldsymbol{\mu}}_{\mathrm{TR}}^{[\bullet]_t}\right)\right] = \frac{1}{L_t} \sum_{h_t}^{H_t} \hat{p}^{[\bullet]_t} \hat{N}^{[\bullet]_t}, t = 1, \ldots, T, \quad [5.71]$$

$$\mathcal{E}\left(\mathcal{M}_6^{(t)}\right) = \sum_{h_t}^{H_t} \hat{p}^{[\bullet]_t} \mathcal{E}\left[\mathcal{M}_{\hat{\boldsymbol{\Sigma}}_{\hat{N}^{[\bullet]_t}}^{[\bullet]_t}}\left(\mathbf{u}^{[\bullet]_t}, \bar{\mathbf{u}}^{[\bullet]_t}\right)\right] = \frac{L_t - 1}{L_t} \sum_{h_t}^{H_t} \hat{p}^{[\bullet]_t} \hat{N}^{[\bullet]_t}, t = 1, \ldots, T. \quad [5.72]$$

Note that the expected value expressions use arrays defined for total returns and not net returns. Covariances for total returns and net returns are the same since a net return vector is obtained by subtracting constant slippage factors from each corresponding total return vector. In addition, slippage factors cancel out in net return difference vectors (see equation [5.57]) since each argument vector includes the constant slippage factors. Equations [5.70] through [5.72] are, therefore, valid for both total and net returns.

Average values for net return, denoted by $\mathcal{A}\left(\mathcal{M}_j^{(t)}\right)$, $j = 4, 5, 6$, for $t = 1, \ldots, T$, are determined as:

$$\mathcal{A}\left(\mathcal{M}_4^{(t)}\right) = \sum_{h_t}^{H_t} \hat{p}^{[\bullet]_t} \left[\sum_{l_t=1}^{L_t} p_{l_t}^{[\bullet]_t} \left(\mathbf{u}_{l_t}^{[\bullet]_t} - \hat{\boldsymbol{\mu}}_{\mathrm{TR}}^{[\bullet]_t}\right)' \left(\hat{\boldsymbol{\Sigma}}_{\hat{N}^{[\bullet]_t}}^{[\bullet]_t}\right)^{-1} \left(\mathbf{u}_{l_t}^{[\bullet]_t} - \hat{\boldsymbol{\mu}}_{\mathrm{TR}}^{[\bullet]_t}\right)\right], t = 1, \ldots, T, \quad [5.73]$$

$$\mathcal{A}\left(\mathcal{M}_5^{(t)}\right) = \sum_{h_t}^{H_t} \hat{p}^{[\bullet]_t} \left[\left(\bar{\mathbf{u}}^{[\bullet]_t} - \hat{\boldsymbol{\mu}}_{\mathrm{TR}}^{[\bullet]_t} \right)' \left(\hat{\boldsymbol{\Sigma}}_{\hat{N}^{[\bullet]_t}}^{[\bullet]_t} \right)^{-1} \left(\bar{\mathbf{u}}^{[\bullet]_t} - \hat{\boldsymbol{\mu}}_{\mathrm{TR}}^{[\bullet]_t} \right) \right], \qquad t = 1, \ldots, T, \tag{5.74}$$

$$\mathcal{A}\left(\mathcal{M}_6^{(t)}\right) = \sum_{h_t}^{H_t} \hat{p}^{[\bullet]_t} \left[\sum_{l_t=1}^{L_t} p_{l_t}^{[\bullet]_t} \left(\mathbf{u}_{l_t}^{[\bullet]_t} - \bar{\mathbf{u}}^{[\bullet]_t} \right)' \left(\hat{\boldsymbol{\Sigma}}_{\hat{N}^{[\bullet]_t}}^{[\bullet]_t} \right)^{-1} \left(\mathbf{u}_{l_t}^{[\bullet]_t} - \bar{\mathbf{u}}^{[\bullet]_t} \right) \right], t = 1, \ldots, T. \tag{5.75}$$

Equations [5.73] through [5.75] are also valid for both total and net returns.

Sample Effectiveness Measures

Values defined by equations [5.64] through [5.69] for returns and equations [5.70] through [5.75] are then used to defined the following sample effectiveness measures:

$$E_j^{(t)} = 100 \frac{\left| \mathcal{A}\left(\mathcal{M}_j^{(t)}\right) - \mathcal{E}\left(\mathcal{M}_j^{(t)}\right) \right|}{\mathcal{E}\left(\mathcal{M}_j^{(t)}\right)}, j = 1, \ldots, 6, t = 1, \ldots, T. \qquad ([5.76])$$

Random samples that approximate a parent distribution reasonably well should yield small $E_j^{(t)}$ values since each of these measures represents a percentage absolute difference between an expected population MSD and an average sample MSD. Empirical results for model **MIMPSLP** are reported in the next chapter.

Chapter 6
Model MIMPSLP Results and Analyses

Analyses of the results obtained by applying the decomposition procedures developed and described in Chapter 4 to the market investment model, **MIMPSLP**, formulated in Chapter 5 are provided in this chapter. Two-stage decomposition of model **MIMPSLP** problem instances based upon the block-separability property is compared with nested and myopic decomposition of those problem instances. Results and analyses for measuring the effectiveness of the underlying random samples and for model applications are also given. All results were obtained on a Pentium II[4] 400 megahertz machine with 384 megabytes of RAM running under Window NT 4.0[5] with service pack six installed.

Material in this chapter is organized into seven sections. Constant data values and representative problem instances are described in the first section. Implementation strategies for two-stage and nested Dantzig-Wolfe and L-Shaped decomposition are given in the second section. Dantzig-Wolfe and L-Shaped decomposition results are analyzed for single period problems in the third section and for multiple period problems in the fourth section. Myopic decomposition results are analyzed and compared with Dantzig-Wolfe and L-Shaped decomposition results in the fifth section. Section six contains results and analyses for measuring sample effectiveness with the Mahalanobis squared distance mea-

[4] Pentium II is a registered trade mark of INTEL Corporation, Santa Clara, CA

[5] Windows NT is a registered trademark of Microsoft Corporation, Redmond, WA

sures developed in Section 5.6.4. Managerial insight through application is the topic of the seventh and final section.

See the two-page Table 5.1 starting on page 180 and Table 5.2 on page 185 for a review of the notation defined for the market investment model.

6.1 Constant Data and Problem Instances

An infinite number of model **MIMPSLP** problem instances can be constructed. A representative sample of these instances is used as the basis for the material in this chapter. The representative sample is built under two conditions:

1. certain problem parameter data is kept constant for all problem instances, and
2. limits are placed upon the number of periods in a planning horizon and on the number of outcomes that may be generated for any period.

Constant problem data is described first, followed by a description of the sample of problem instances constructed based upon the two conditions above.

6.1.1 Constant Data

Fixed transaction costs, or slippage, data (see Section 5.1.1 starting on page 165) and scenario generation distribution data (see Section 5.6 starting on page 207) are kept constant for all problem instances. These data are controlled by data files that are input to the scenario generator code library described in Section 5.6. The data files utilized for the problem instances described below are applicable to set of $N = 68$ securities. Slippage is

approximated with 4 piece-wise linear segments in each of the two slippage regions so that $K = 4 + 4 = 8$. Piece-wise linear approximation break points and slopes are assigned to minimize average absolute error in the first slippage region and to limit the maximum relative error to ten percent in the second slippage region. Only the linear deviations model is examined so that deviation is modeled exactly with a single linear function ($Q = 1$). All problem instances are based upon zero position holdings ($\mathbf{y}^{(0)} = \mathbf{0}$) at the beginning of the planning horizon. Positions for each security are not bounded from above ($\mathbf{Y}_{\bullet t}^{\max} = \infty$) and no short positions are allowed ($\mathbf{Y}_{\bullet t}^{\min} = 0$) during any period, t, in the planning horizon.

Problem instances described in the next subsection span planning horizons of from one to five periods. Each period, t, will have a length, $\pounds_t$, in days selected from six possible values. The risk aversion factor, λ_t, capital discounting factor, ϱ_t, and risk budget amount, β_t, are also held constant for periods of the same length. Selected period lengths and the dependent problem parameters are described in Table 6.1. Constant discount factors of 1.0 (no discounting of capital) given in Table 6.1 are considered reasonable given the relatively short period lengths. The risk aversion factors shown in Table 6.1 result in very low aversion to risk (almost no slack in the risk budget constraints) except when the applicable period is the first in a multiple period planning horizon. Risk aversion for the first period in the latter case is considered fairly high to moderate (30 to 50 percent slack in the risk budget constraint).

Table 6.1: Period Lengths and Dependent Data

Period Length[a] $\pounds_t$	**Risk Aversion** λ_t	**Discount Factor** ϱ_t	**Scaled Budget** $\tilde{\beta}_t$[b]	**Actual Budget** β_t[c]
2	0.0334	1.0	2,000,000.00	2,828,427.12
3	0.0784	1.0	2,000,000.00	3,464,101.62
5	0.2160	1.0	2,000,000.00	4,472,135.95
10	0.5620	1.0	2,000,000.00	6,324,555.32
15	0.8350	1.0	2,000,000.00	7,745,966.69
25	1.2520	1.0	2,000,000.00	10,000,000.00

[a] In days.

[b] In dollars (US) – the notation $\tilde{\beta}_t$ is new

[c] In dollars (US) – actual budget is: $\beta_t = \tilde{\beta}_t\sqrt{\pounds_t}$

6.1.2 Problem Instances

Problem instances are constructed for planning horizons with from one to five periods. Instances for horizons with a single period and those with two or more periods are described separately below.

Single Period Problem Instances

Planning horizons with a single period give considerable insight into the application of decomposition techniques. Fairly large single period problems can be solved using both the grand LP formulation and decomposition procedures in relative short amounts of time. Most multiple period problem instances considered in this chapter cannot, however, be solved in the grand LP formulation due to insufficient computer memory resources. Therefore, single period problems provide a better platform than multiple period problems for comparing the efficiency of decomposition methods with grand LP solutions. Single period problems are also critical components in the most efficient decomposition strategies discussed in the next section.

All planning horizons with a single period that are considered below have period (horizon) lengths of 5 days. Only the number of random outcomes generated during the period differentiates single period problem instances. Specific instances are given with applicable results and analyses in Section 6.3 below.

Table 6.2: Scenario Generation Data for Multiple Period Problem Instances

Number Periods T	Periods' Lengths $[\pounds_t]$	Outcomes by Problem Size and Number of Periods		
		Small 10^4 Scenarios	Medium 10^5 Scenarios	Large 10^6 Scenarios
2	[5 25]	[100 100]	[500 200]	[1000 1000]
3	[5 10 15]	[50 20 10]	[80 50 25]	[100 100 100]
4	[2 3 10 15]	[25 20 10 2]	[40 25 20 5]	[80 50 25 10]
5	[2 3 5 10 10]	[20 10 5 5 2]	[25 20 10 10 2]	[50 25 20 10 4]

Multiple Period Problem Instances

Multiple period problem instances have planning horizons of from 2 to 5 periods. Individual problems are generated with 10^4, 10^5, or 10^6 scenarios and are classified as small, medium, or large respectively. Period lengths and the numbers of outcomes by period are shown in Table 6.2 based upon the number of periods, T, in the planning horizon.

Problem instances are labelled as **P*T*-Small**, **P*T*-Medium**, or **P*T*-Large** where T is replaced with the number of periods in the planning horizon. Problems are copied to CPLEX using nine arrays to transport problem data. The copy process can require a considerable amount of memory since CPLEX creates its own memory block (see function *CPXcopylp* in the CPLEX reference manual [109]) doubling the amount of memory required to copy problem data. Memory allocated for the original problem data arrays cannot be freed until the copy process has completed. Problem size information based upon the formulae in Table 5.3 on page 204 and the memory required for the copy process is given in Table 6.3 for each of the 12 multiple period problem instances.

The entries in Table 6.3 demonstrate that model **MIMPSLP** problems can be extremely large. Over 23 gigabytes of RAM would be required to copy the primal grand LP

Table 6.3: Sizes of Multiple Period Problem Instances (Primal Grand LP Formulation)

Problem Label	Number of				Copy Memory[c]	Size Rank[d]
	Nodes[a]	Constraints	Variables	Non-zeros[b]		
P2-Small	101	23,937	71,912	840,992	12.6	12
P2-Medium	501	169,137	407,112	7,649,792	106.3	10
P2-Large	1,001	1,138,137	1,613,612	70,498,292	907.9	5
P3-Small	1,051	155,037	654,262	2,263,142	49.5	11
P3-Medium	4,081	663,177	2,601,652	13,009,052	245.3	9
P3-Large	10,101	2,393,937	7,191,912	84,120,992	1,261.1	4
P4-Small	5,526	772,587	3,397,437	8,962,217	223.2	8
P4-Medium	21,041	3,003,657	12,998,132	38,398,172	903.8	6
P4-Large	104,081	15,363,177	64,801,652	224,809,052	4,909.6	2
P5-Small	6,221	868,497	3,823,472	10,002,632	250.2	7
P5-Medium	55,526	7,762,587	34,137,437	90,022,217	2,242.5	3
P5-Large	276,301	39,129,537	170,372,512	482,622,392	11,595.6	1

[a] Cumulative nodes in all periods

[b] In the technology matrix

[c] Memory in megabytes required to copy to CPLEX – effectively doubled during copy process

[d] By descending order of copy memory required – rank 1 requires the most memory

formulation of problem **P5-Large** to CPLEX. Large scale problems are a characteristic of most stochastic linear programs. The size rank indices (last column) in Table 6.3 will be used to characterize the relative size of the 12 problem instances, e.g. **P5-Large** is the largest while **P2-Small** is the smallest.

6.2 DWD and LSD Implementation Strategies

Implementation strategies for Dantzig-Wolfe and L-Shaped decomposition are defined using the major strategy, minor strategy, and tactics planning scheme described in Section 4.4.3. Seven major strategies are available in the model *MIMPSLP* library. Major strategy levels 1 through 6 are described in the first three subsections below. Only two-stage decomposition based upon the block-separable property of model **MIMPSLP** problems is used under major strategies 1 through 6. Major strategy 7 is defined in the fourth subsection. Both two-stage and nested decomposition procedures are possible under major strategy 7. Preliminary decomposition results are discussed in the fifth and final subsection.

Tables are used in the first three subsections to define the minor strategies and tactics levels for the applicable major strategies. The term *eligible* is used in these tables to describe decomposition subproblems that may be used to generate a new optimality cut (LSD) or extreme point activity (DWD). An eligible DWD subproblem yields an extreme point with negative reduced cost. An eligible LSD subproblem has a relaxation function value that is less than the corresponding relaxation variable value. The subproblem yield-

ing the most negative reduced cost (DWD)/relaxation function minus relaxation variable difference (LSD) during any iteration is said to be the *most eligible* subproblem.

6.2.1 Major Strategies 1 and 2

Major strategies 1 and 2 may be used with problems having an arbitrary number of periods. The initial RMP is formed with the primal budget constraints (LSD) from all nodes in the decision tree or with the dual variables to these constraints (DWD). Major strategies 1 and 2 are the same except for the values assigned to the right-hand-sides (LSD)/objective function coefficients (DWD) of the initial optimality cuts (LSD)/convexity activities (DWD). Major strategy 1 assigns zero to each of these quantities (see Section E.1.1). Major strategy 2 uses the feasible solution returned from the first half-cycle of the duals lead version of the myopic dual-primal cycling algorithm (see section 4.5 and Appendix B) to assign these values. The minor strategies and tactics that are available for the first two major strategy levels are shown in Table 6.4.

6.2.2 Major Strategies 3 and 4

Major strategies 3 and 4 may only be used with problems that have two or more periods. The initial RMP is formed with the primal budget constraints (LSD) from the nodes in all periods except the last or with the dual variables to these constraints (DWD). Major strategies 1 and 2 are the same except for the manner in which the subproblems associated with nodes in the terminal period are copied to CPLEX. Each nodal component subproblem (see

Table 6.4: Minor Strategies (Minor) and Tactics for Major Strategies 1 and 2

Minor Value	Minor Description	Tactics Value	Tactics Description
1	1 SUB for all slippage/deviation	1	Add if eligible [1][a]
2	1 SUB for all slippage and 1 SUB for all deviation	1	Add most eligible [1]
		2	Add all eligible [2]
3	1 SUB for slippage for each period and 1 SUB for deviation for each period	1	Add most eligible [1]
		2	Add most eligible among slippage and most eligible among deviation [2]
		3	Add all eligible among slippage and most eligible among deviation $[T+1]$
		4	Add all eligible $[2T]$
4[b]	1 SUB for slippage at each node and 1 SUB for deviation at each node	1	Add most eligible [1]
		2	Add most eligible among slippage and most eligible among deviation [2]
		3	Add most eligible among slippage during each period and most eligible among deviation $[T+1]$
		4	Add most eligible among slippage during each period and most eligible among deviation during each period $[2T]$
		5	Add all eligible among slippage and most eligible among deviation $[H^{(T)}+1]$
		6	Add all eligible among slippage and most eligible among deviation during each period $[H^{(T)}+T]$
		7	Add all eligible $[2H^{(T)}]$
4[c]	1 SUB for slippage at each security and 1 SUB for deviation for each of $G=1$ or U[d] outcomes' groupings	1	Add most eligible [1]
		2	Add most eligible among slippage and add deviation if eligible [2]
		3	Add most all eligible among slippage and add deviation if eligible $[N+1]$
		≥ 4	Add all eligible $[N+U]$

[a] [A]: A is the maximum number of constraints/activities that can be added at any iteration

[b] For problems over 2 or more periods only

[c] For problems over a single period only

[d] U is the Minor Strategy value when this value is greater than or equal to 4

Appendix E) is copied in whole to CPLEX under major strategy 3. Each subproblem in major strategy 3 is then solved with a *cold start* (no advance basis). Only the first nodal component subproblem, node $(T, 1)$, is copied in whole under major strategy 4. Subproblems at the remaining terminal period nodes are created by changing only CPLEX problem data that differs from the previous subproblem. The latter procedure is fairly efficient since nodal component subproblems in the same period are the same size and share a significant amount of common data. Major strategy 4 then uses a cold start only on the first nodal component subproblem while *warm starts* (with an advance basis) are used with the remaining subproblems. The minor strategies and tactics that are available for major strategies 3 and 4 are shown in Table 6.5.

6.2.3 Major Strategies 5 and 6

Major strategies 5 and 6 are the same as major strategies 3 and 4 except the final period nodal component subproblems are not solved during every decomposition iteration. Instead, a greedy procedure (see Section 3.3.6 starting on page 114) is used in which these subproblems are solved intermittently. The minor strategies and tactics that are available for major strategies 5 and 6 are shown in Table 6.6.

6.2.4 Major Strategy 7

Major strategy 7 may be used only with problems that have two or more periods. This major strategy is similar to major strategy 5 except that nodal component subproblems may

Table 6.5: Minor Strategies (Minor) and Tactics for Major Strategies 3 and 4

Minor Value	Minor Description	Tactics Value	Tactics Description
1	1 SUB for all	1	Add if eligible [1][a]
2	1 SUB for slippage/FPN[b] and 1 SUB for all deviation	1	Add most eligible [1]
		2	Add all eligible [2]
3	1 SUB for slippage for each $t < T$, 1 SUB for FPN, and 1 SUB for deviation for each $t < T$	1	Add most eligible [1]
		2	Add most eligible among slippage/FPN and most eligible among deviation [2]
		3	Add all eligible among slippage/FPN and most eligible among deviation $[T + 1]$
		4	Add all eligible $[2T - 1]$
4	1 SUB for slippage at each node in each $t < T$, 1 SUB for each FPN, and 1 SUB for deviation at each node in each $t < T$	1	Add most eligible [1]
		2	Add most eligible among slippage/FPN and most eligible among deviation [2]
		3	Add most eligible among slippage during each $t < T$, most eligible among FPN, and most eligible among deviation $[T + 1]$
		4	Add most eligible among slippage during each $t < T$, most eligible among FPN, and most eligible among deviation during each $t < T$ $[2T - 1]$
		5	Add all eligible among slippage/FPN and most eligible among deviation $[H^{(T)} + 1]$
		6	Add all eligible among slippage/FPN and most eligible among deviation during each $t < T$ $[H^{(T)} + T - 1]$
		7	Add all eligible $[H^{(T-1)} + H^{T}]$
5	1 SUB for slippage at each security at each node in each $t < T$, 1 SUB for each FPN, and 1 SUB for deviation for each node in each $t < T$	1	Add all eligible $[NH^{(T-1)} + H^{(T)}]$

[a] $[A]$: A is the maximum number of constraints/activities that can be added at any iteration

[b] FPN stands for final period nodes

Table 6.6: Minor Strategies (Minor) and Tactics for Major Strategies 5 and 6

Minor Value	Minor Description	Tactics Value	Tactics Description
3[a]	1 SUB for slippage for each $t < T$, 1 SUB for FPN[c], and 1 SUB for deviation for each $t < T$	1	Add most eligible [1][b]
		2	Add most eligible among slippage/FPN and most eligible among deviation [2]
		3	Add all eligible among slippage/FPN and most eligible among deviation $[T + 1]$
		4	Add all eligible $[2T - 1]$
4	1 SUB for slippage at each node in each $t < T$, 1 SUB for each FPN, and 1 SUB for deviation at each node in each $t < T$	1	Add most eligible [1]
		2	Add most eligible among slippage/FPN and most eligible among deviation [2]
		3	Add most eligible among slippage during each $t < T$, most eligible among FPN, and most eligible among deviation $[T + 1]$
		4	Add most eligible among slippage during each $t < T$, most eligible among FPN, and most eligible among deviation during each $t < T$ $[2T - 1]$
		5	Add all eligible among slippage/FPN and most eligible among deviation $[H^{(T)} + 1]$
		6	Add all eligible among slippage/FPN and most eligible among deviation during each $t < T$ $[H^{(T)} + T - 1]$
		7	Add all eligible $[H^{(T-1)} + H^{(T)}]$
5	1 SUB for slippage at each security at each node in each $t < T$, 1 SUB for each FPN, and 1 SUB for deviation for each node in each $t < T$	1	Add all eligible $[NH^{(T-1)} + H^{(T)}]$

[a] Minimum allowed Minor Strategy is 3 to be consistent with other Major Strategies

[b] $[A]$: A is the maximum number of constraints/activities that can be added at any iteration

[c] FPN stands for final period nodes

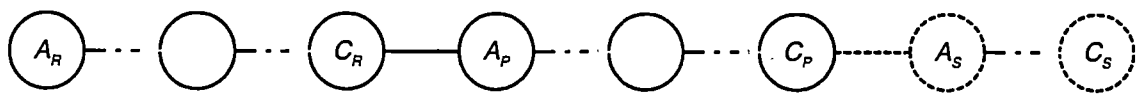
be created for nodes in periods prior to the last but later than the first. The resulting nodal component subproblems may be solved in the grand LP formulation (two-stage decomposition) or with a recursive application of major strategy 7 (nested decomposition). Cold starts are used for all grand LP solutions – there is no complementary warm start major strategy for reasons given in the next subsection.

Risk budget constraints in periods $A_R = 1$ through $C_R = \bar{t}$ are used to form the initial RMP for a specified $\bar{t}$, referred to as the RMP *cutoff* period, such that $1 \leq \bar{t} < T$ (also see Section 4.4.3 starting on page 146). Note that major strategy 7 is the same as major strategy 5 if the RMP cutoff period is one less than the terminal period. The RMP cutoff period is set equal to the value of the minor strategy.

Two-stage decomposition is applied if the value of tactics is between 1 and 4 inclusive and nested decomposition is applied if the value of tactics is between 5 and 12 inclusive. The *primary subproblem anchor period* is defined as $A_P = \hat{t} = \bar{t} + 1$ and the primary subproblem cutoff period, C_P, is defined as

$$C_P = \begin{cases} T, & \text{if } 1 \leq \text{tactics} \leq 4, \\ \left\lfloor \frac{T-\bar{t}}{2} \right\rfloor + \bar{t}, & \text{if } 5 \leq \text{tactics} \leq 8, \\ T-1, & \text{if } 9 \leq \text{tactics} \leq 12. \end{cases}$$

If the tactics value is between 5 and 12 inclusive, the *secondary subproblem* anchor period, A_S, and cutoff period, C_S, are defined as $A_S = C_P + 1$ and $C_S = T$. Figure 6.1 summarizes these concepts.



LEGEND

A_R : Anchor period for the RMP
C_R : Cutoff period for the RMP
A_P : Anchor period for the primary SUBs
C_P : Cuttoff period for the primary SUBs
A_S : Anchor period for the secondary SUBs (nested tactics only)
C_S : Cutoff period for the secondary SUBs (nested tactics only)

RMP - All Tactics

$A_R = 1$
C_R = Minor Strategy ($\bar{t}$)
$A_P = C_R + 1$

SUBs

Tactics	Primary SUB Cutoff	Solution - Secondary SUB Cutoff
1 - 4	$C_P = T$	Grand LP
5 - 8	$C_P = \left\lfloor \frac{T - \bar{t}}{2} \right\rfloor + \bar{t}$	Nested Decompostion With $A_S = C_P + 1$, $C_S = T$
9 - 12	$C_P = T - 1$	

Figure 6.1: Major Strategy 7 Schemetic and Description

The value of tactics also determines the problem formulation-simplex solver combination that will be applied to the grand LP for applicable nodal component subproblems – primary subproblems when $1 \leq$ tactics ≤ 4 or secondary subproblems when $5 \leq$ tactics ≤ 12. Four combinations are possible and are shown in Table 6.7 relative to the tactics value.

Table 6.7: Major Strategy 7 Subproblem Formulation-Simplex Solver Combinations

Tactics Value	Formulation-Solver	Shorthand Notation
1, 5, or 9	Primal Formulation - Primal Simplex	(PF-PS)
2, 6, or 10	Primal Formulation - Dual Simplex	(PF-DS)
3, 7, or 11	Dual Formulation - Primal Simplex	(DF-PS)
4, 8, or 12	Dual Formulation - Dual Simplex	(DF-DS)

6.2.5 Preliminary Decomposition Results

The major strategy controls the amount of problem information that is used to construct the initial RMP and, therefore, dictates how much of the problem structure must be accounted for by subproblems. In general, increasing major strategy levels lead to smaller initial relaxed/restricted master problems and to more or larger subproblems. Dantzig-Wolfe and L-Shaped decomposition routines were developed and implemented in the sequential order of the major strategy indices. Routines based upon major strategy 7 were developed last using information collected from studies of decomposition results obtained with major strategies 1 through 6. Similarly, the development of routines based upon major strategies

5 and 6 (3 and 4) was motivated by the results obtained with routines based upon major strategies 1 through 4 (1 and 2).

Minor strategy and tactics indices within a given major strategy level (1 through 6) determine the detail of the information used from one decomposition iteration to modify the RMP for the next iteration. Smaller minor strategy indices yield fewer, fairly large aggregated subproblems while larger minor strategy levels result in separation of aggregated subproblems into more, smaller subproblems. The maximum number of optimality cuts or extreme point activities that may be added to the RMP increases with the tactics level within a major-minor strategy combination.

Studies of the results obtained with routines based upon major strategy levels 1 through 6 inspired the eight *rules of thumb* described in the following list. Applicability of these rules is effected by such factors as available computer memory, efficiency of the simplex solver that is employed, and the structure of the stochastic program under study. The rules are considered, however, to provide a good basis for the types of issues that should be addressed in order to construct an efficient solution algorithm for a general stochastic linear program.

1. A small initial RMP is more efficient for larger problems (hundreds or more nodes) while a larger initial RMP is better for smaller problems (tens of nodes or less). Note that the size of the initial RMP refers to the amount of problem information used to construct the RMP – a *small* initial RMP for a large problem may be significantly larger than a *large* initial RMP for a smaller problem.

2. More, smaller subproblems are more efficient than fewer, more aggregated subproblems. This rule is especially important if separation of large subproblems yields smaller, more easily solved subproblems.

3. Addition of multiple cuts/activities to the RMP during an iteration is more effective than using fewer cuts/activities. More problem information is added to the RMP with multiple cuts/activities than can be added with fewer, more aggregated cuts/activities.

4. No significant advantage is gained by removal of inactive cuts (not tight) or activities (not basic). RMP solution times are more dependent on the number of additional cuts/activities than on the size of the RMP since all solutions after the first iteration use an advanced basis. Any advantages gained by increasing the available active memory through removal of cuts/activities is, in general, more than offset by the overhead involved in the removal process. This rule is, of course, effected by the amount of available computer memory.

5. No advantage is gained by using the myopic dual-primal cycling algorithm to generate initial optimality cuts/extreme points. The simple initialization procedures described in Appendix E require significantly less time than initialization procedures based upon the MDPCA. Moreover, the two initialization methods generally do not yield a significant difference in the number of decomposition iterations required to achieve a specified gap tolerance.

6. Solving grand LP formulated subproblems with cold starts is generally as or more effective than solving these subproblems with warm starts. Information obtained with the CPLEX preprocessing algorithm (see the CPLEX user's manual [110, pages 72 - 73]) applied during a cold start solution procedure appears to more than offset any advantages gained by using advance basis information.

7. Grand LP based solutions can be obtained more efficiently than decomposition based solutions when the underlying problems are relatively small. The simplex solver that is employed can greatly influence the applicability of this rule.

8. Greedy algorithms are more efficient than algorithms that solve every subproblem during each decomposition iteration. Significant time savings with relative little loss of information is realized when larger, more complicated subproblems (generally embedded deep in the decision tree) are solved on an intermittent basis.

The eight rules of thumb motivated development of solution routines based upon major strategy 7. These routines proved to be significantly more efficient on problems with two or more periods than routines based upon the smaller major strategy levels. Consequently, all results for problems with multiple periods that are reported in the next section were obtained using major strategy 7.

6.3 DWD and LSD Results – Single Period Problems

Thirteen single period problems that differ only the number of outcomes generated during the period are examined. The number of outcomes vary from a low of 2 to a high of 100,000. Table 6.8 lists the thirteen problems by size (number of outcomes L) and shows the times required to obtain grand LP solutions. Times are given for each of the four possible formulation-solver combinations using the shorthand notation of Table 6.7 on page 240. Note that the primal formulation is best for only the five smallest problems while the dual formulation is best for the larger problems. Data in Table 6.8 also indicates that there can be large variations in solution time relative to the choice of a simplex solver. The overall effect can be drastic – the slowest time for the largest problem is almost 25 times greater than the fastest time for that problem. Figure 6.2 illustrates the fastest grand LP solution times relative to the size ranking of the 13 single period problems.

Each of the 13 single period problems was also solved with Dantzig-Wolfe and L-Shaped decomposition. Only major strategy 1 applies to single period problems and minor strategy 4 for single period problems (see Table 6.4 on page 234) is the best applicable minor strategy. Each problem was solved under nine different tactics values to determine the effect of varying the number of subproblems. The Dantzig-Wolfe decomposition restricted master problem was solved with the dual formulation-primal simplex combination at each iteration while the primal formulation-dual simplex combination was applied to the relaxed master problem of L-Shaped decomposition. These two formulation-solver combi-

Table 6.8: Single Period Grand LP Solution CPU Times (Minutes)

Outcomes L	**Formulation-Solver Combination**[a]			
	(PF-PS)	**(PF-DS)**	**(DF-PS)**	**(DF-DS)**
2	**0.0005**[b]	**0.0005**	0.0008	0.0008
10	0.0010	**0.0008**	0.0013	0.0013
20	0.0018	**0.0010**	0.0022	0.0018
50	0.0030	**0.0017**	0.0072	0.0023
100	0.0062	**0.0033**	0.0103	**0.0033**
500	0.0604	0.0250	0.0616	**0.0113**
1,000	0.2333	0.0988	0.1611	**0.0324**
2,500	1.330	0.6848	0.5873	**0.2140**
5,000	4.830	3.700	1.890	**0.8796**
10,000	9.340	8.120	3.750	**3.270**
20,000	42.41	32.53	**8.450**	9.930
50,000	235.7	204.6	**32.48**	49.86
100,000	1,803	1,525	**72.59**	111.4

[a] Using the shorthand notation defined in Table 6.7 on page 240

[b] Bold face times in each row are the fastest times for the individual problem

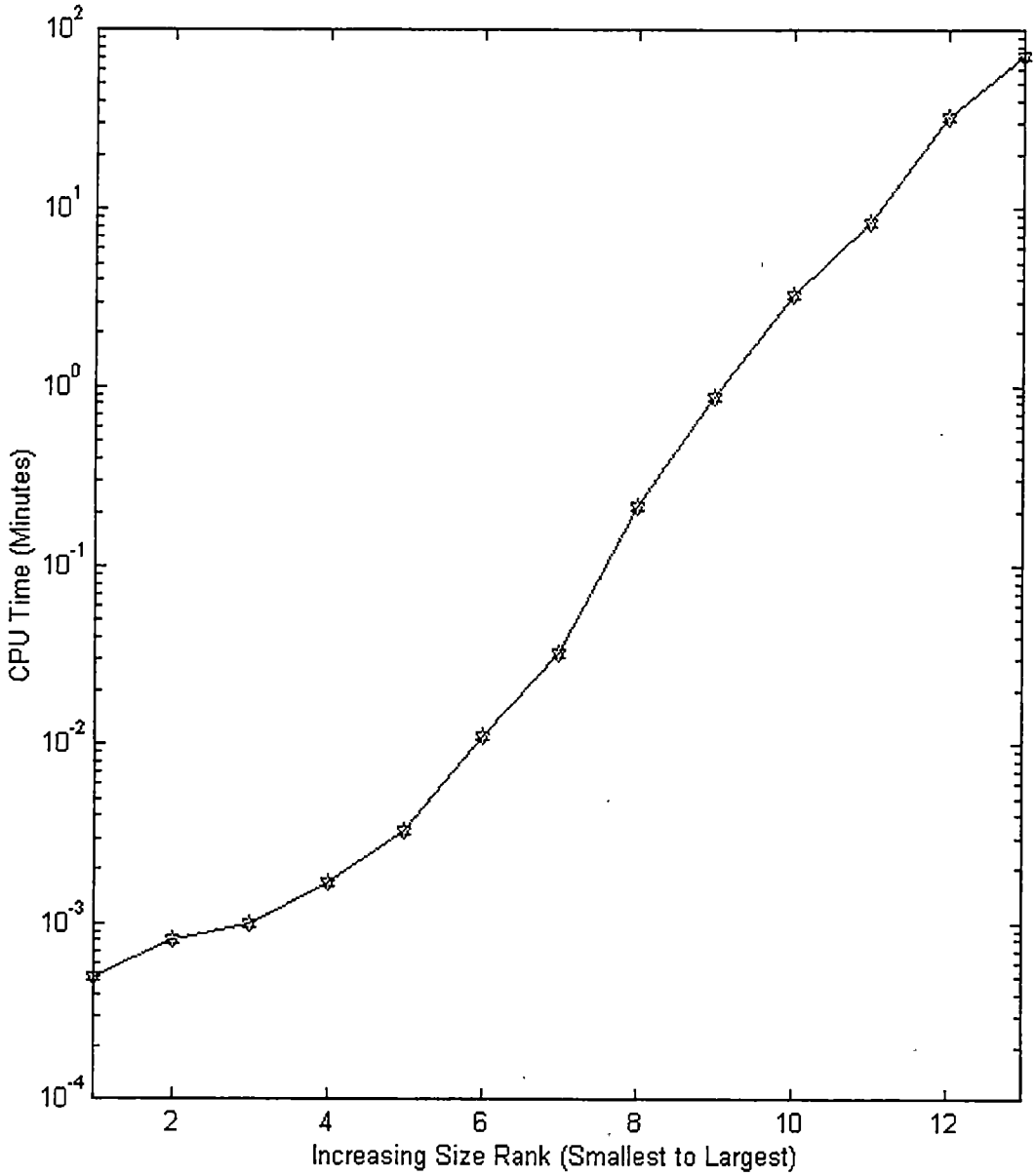


Figure 6.2: Fastest Grand LP Single Period Solution Times

nation versus type RMP matches proved to be the best in all model **MIMPSLP** problems that were examined. All decomposition procedures for the single period problems employed an acceptable termination relative tolerance of 5×10^{-8}:

$$\frac{UpperBound - LowerBound}{LowerBound} < 5 \times 10^{-8}.$$

Solutions satisfying the above termination tolerance are considered exact since the default value for all CPLEX LP simplex solver tolerances is 10^{-6} (see the CPLEX user's manual [110, pages 85 - 86]) and the default tolerances were used in all calls to CPLEX. Table 6.9 contains the solution times required for the Dantzig-Wolfe and L-Shaped decomposition procedures.

The solution times in Table 6.9 indicate that each decomposition method-problem combination has an optimum number of subproblems. Therefore, increasing the number, and thus decreasing the size, of subproblems is beneficial to some threshold beyond which this process becomes detrimental to the decomposition algorithm. Note also that data in Table 6.9 show that Dantzig-Wolfe decomposition consistently outperforms L-Shaped decomposition.

The CPU times in Tables 6.8 and 6.9 demonstrate that obtaining solutions with the grand LP formulation is more efficient for problems with less than about 1,000 outcomes while either decomposition method is more efficient for larger problems. Table 6.10 contains the ratios for the fastest grand LP solution times to the fastest Dantzig-Wolfe and L-Shaped decomposition times and for the fastest DWD times to the fastest LSD times.

Table 6.9: Single Period DWD/LSD Solution CPU Times (Seconds)

Outcomes	DWD(1.4.U)/LSD(1.4.U)[a] for U[b]:								
L	**1**	**5**	**10**	**20**	**25**	**50**	**75**	**100**	**125**
2	**0.0921**[c]	–[d]	–	–	–	–	–	–	–
	0.1202	–	–	–	–	–	–	–	–
10	0.0962	**0.0781**	0.0801	–	–	–	–	–	–
	0.1202	**0.1042**	0.1262	–	–	–	–	–	–
20	0.1662	0.1442	**0.1402**	0.1522	–	–	–	–	–
	0.2464	**0.2383**	0.2424	0.2363	–	–	–	–	–
50	0.2283	0.2002	**0.1723**	0.1863	0.1903	0.1943	–	–	–
	0.3405	**0.2403**	0.3805	0.3205	0.3405	0.2904	–	–	–
100	0.3805	0.3004	**0.2403**	0.2804	0.3004	0.3104	0.3004	0.3805	–
	0.7811	0.5708	0.5107	0.5608	**0.4807**	0.5708	0.5608	0.4907	–
500	0.8813	**0.5708**	0.6810	0.7010	0.6810	1.052	1.222	1.512	1.652
	1.502	**1.142**	1.212	1.322	1.592	2.233	2.534	2.784	2.664
1,000	1.823	**0.9614**	1.022	1.001	1.242	1.723	2.434	2.544	2.954
	2.774	**1.783**	2.113	2.233	2.694	3.655	5.107	6.089	7.351
2,500	5.838	2.764	**2.163**	2.363	2.233	3.365	3.755	4.787	5.718
	7.521	4.216	**3.705**	4.607	4.527	7.391	9.243	11.76	14.65
5,000	11.72	5.198	4.356	4.166	**3.935**	5.027	5.548	7.731	8.562
	13.67	7.230	**6.820**	7.521	7.010	10.13	12.65	17.19	19.52
10,000	19.40	10.32	7.611	**6.630**	7.541	8.082	9.434	11.61	12.62
	20.88	12.87	10.03	**10.01**	11.86	15.42	19.31	26.05	28.14
20,000	27.98	17.14	15.35	13.51	**12.76**	13.60	16.04	18.38	18.45
	28.53	18.89	18.30	**18.17**	18.73	23.87	30.06	38.06	43.59
50,000	92.81	55.82	45.36	38.34	36.37	**34.37**	38.30	43.28	39.70
	93.40	57.93	49.99	45.78	**45.42**	47.38	60.21	77.6	80.01
100,000	163.7	114.2	96.37	79.56	79.56	72.69	70.45	**67.92**	71.54
	163.0	117.1	102.0	89.13	**88.67**	96.10	101.4	106.9	125.0

[a] DWD solution times are above LSD solution times in each cell

[b] U is the number of outcomes' groups – see Table 6.4 on page 234

[c] Bold face times in each row are the fastest times for the individual problem

[d] Nonapplicable entries are indicated with a dash, –

Table 6.10: Single Period Solution CPU Time Ratios

Outcomes L	Ratios of Fastest Solution Times		
	GLP[a]/DWD	GLP/LSD	LSD/DWD
2	0.3257	0.2496	1.305
10	0.6146	0.4607	1.334
20	0.4280	0.2518	1.700
50	0.5920	0.4245	1.395
100	0.8240	0.4119	2.000
500	1.188	0.5937	2.001
1,000	2.022	1.090	1.855
2,500	5.936	3.466	1.713
5,000	13.41	7.738	1.733
10,000	29.59	19.60	1.510
20,000	39.73	27.90	1.424
50,000	56.70	42.91	1.322
100,000	64.13	49.12	1.306

[a] Grand Linear Program

These ratios clearly demonstrate that decomposition can be a very attractive procedure for solving large scale linear programs. Dantzig-Wolfe decomposition of the largest single period problem is, for example, more than 64 times as fast as the best grand LP solution procedure. Figure 6.3 graphically illustrates the ratios contained in Table 6.10.

6.4 DWD and LSD Results – Multiple Period Problems

Two-stage and nested decomposition procedures were applied to the 12 multiple period problem instances defined in Section 6.1.2 and listed in Table 6.3 on page 231. Major strategy 7 as defined in Section 6.2.4 and illustrated in Figure 6.1 on page 239 was used to define the employed decomposition procedures. Individual procedures are labelled according to the scheme described in Section 4.4.3 starting on page 146 – DWD($i.j.k$)/LSD($i.j.k$) where $i = 7$ is the major strategy, j is the minor strategy, and k is the tactics.

Dantzig-Wolfe decomposition restricted master problems were solved with the dual formulation-primal simplex combination at each iteration while the primal formulation-dual simplex combination was applied to each relaxed master problem of L-Shaped decomposition. All decomposition procedures for each multiple period problem except **P5-Large** employed an acceptable termination relative tolerance of 5×10^{-8}. The decision tree for each problem except **P5-Large** was stored in random access memory. Problem **P5-Large** required special handling due to its size and solution procedures for this problem are described in Section 6.4.3 below.

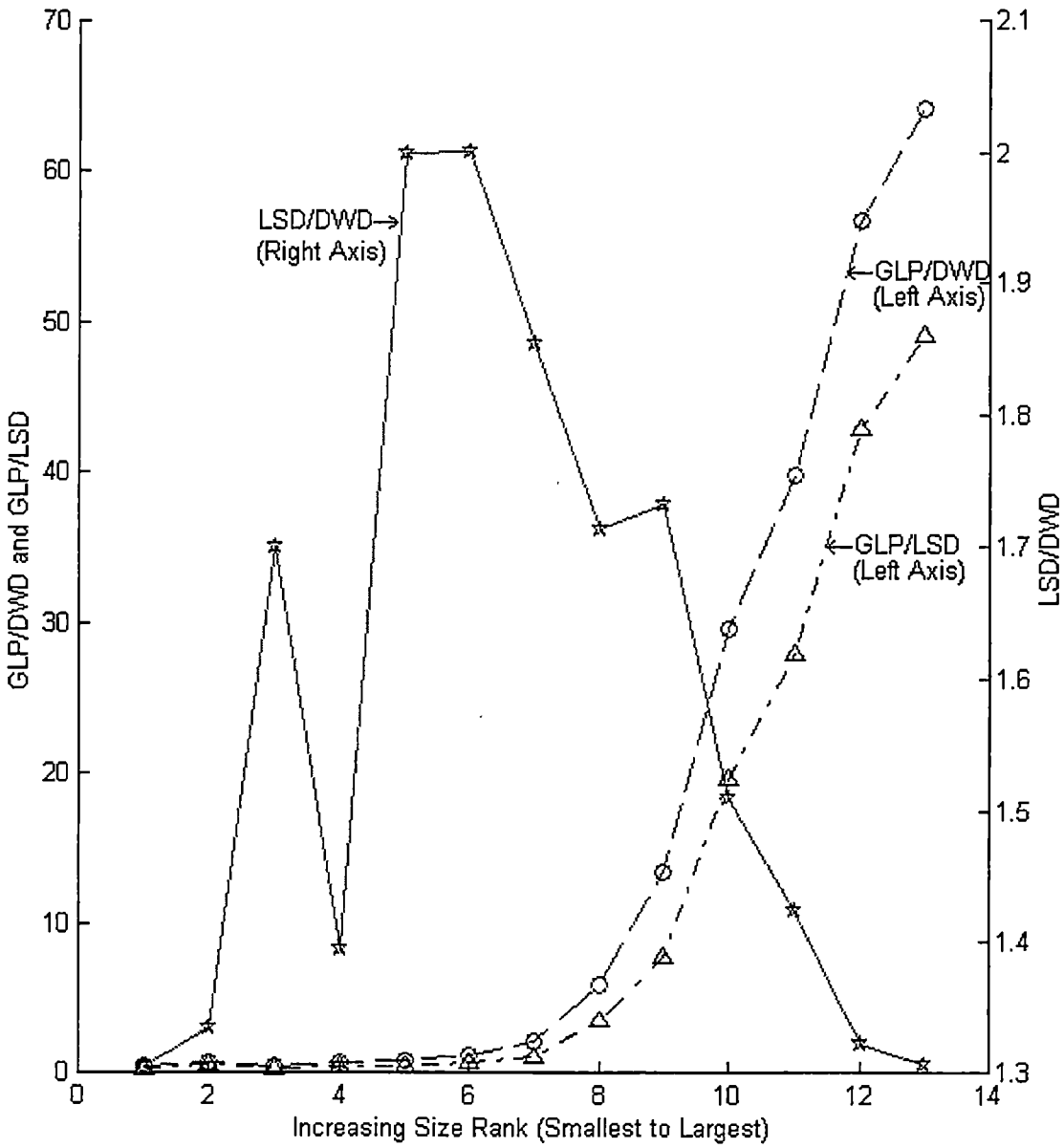


Figure 6.3: Solution CPU Time Ratios for Single Period Problems

An overview of the grand LP and decomposition solution results for the multiple period problems is given in Section 6.4.1. Two-stage and nested decomposition results are analyzed in Sections 6.4.2 and 6.4.3 respectively. Section 6.4.4 contains concluding general comments concerning Dantzig-Wolfe and L-Shaped decomposition.

6.4.1 Overview

Exact solutions (5×10^{-8} relative tolerance) were obtained on all problems except **P5-Large** using two-stage decomposition techniques. Nested decomposition was used to solve problem **P5-Large** to a relative tolerance of 10^{-2}. Grand LP solutions, on the other hand, could be determined for only the three smallest multiple period problems. Available computer memory prevented some problems (*not enough memory*) from even being copied to CPLEX in the grand LP formulation. Some problems could be copied to CPLEX as a grand LP but not solved in a reasonable amount of time due to a drain on the available memory (*insufficient memory*). A problem in the insufficient memory category can be identified by the very low (0 - 20) percentage of CPU usage registered by the Windows NT Task Manager (see the Windows NT resource guide [146, Chapter 11]) after the problem has been copied to CPLEX. The fastest solution times using decomposition and grand LP formulations are shown in Table 6.11.

Problem **P4-Large** is the largest known stochastic linear program to be solved on a computer not in a mainframe or higher class. The largest solved problem found in an extensive literature search has a grand LP formulation of approximately 5 million columns

Table 6.11: Multiple Period Problems' Fastest Solution CPU Times (Minutes)

Problem Label	Grand LP		DWD and LSD		Times Ratio[c]
	Combination[a]	Time	Method(Strategy)	Time[b]	
P2-Small	(DF-DS)	5.9492	DWD(7.1.2)	2.0423	2.9130
P2-Medium	(DF-DS)	226.91	DWD(7.1.2)	43.742	5.1875
P2-Large	Not Enough Memory[d]		LSD(7.1.4)	894.75	NA[e]
P3-Small	(PF-PS)	58.385	DWD(7.2.2)	10.291	5.6734
P3-Medium	Insufficient Memory[f]		DWD(7.2.2)	73.960	NA
P3-Large	Not Enough Memory		DWD(7.2.2)	366.87	NA
P4-Small	Insufficient Memory		DWD(7.2.2)	30.786	NA
P4-Medium	Not Enough Memory		DWD(7.2.2)	492.80	NA
P4-Large	Not Enough Memory		DWD(7.2.2)	4,853.9	NA
P5-Small	Insufficient Memory		DWD(7.3.2)	58.336	NA
P5-Medium	Not Enough Memory		DWD(7.3.2)	608.57	NA
P5-Large	Not Enough Memory		DWD(7.2.10)	4,086.1	NA

[a] Formulation-solver combination – shorthand notation defined in Table 6.7 on page 240

[b] **P5-Large** relative tolerance gap is 10^{-2} – all others use $5x10^{-8}$

[c] Ratio of grand LP time to decomposition time

[d] Not enough memory available to copy problem in grand LP formulation to CPLEX

[e] Not available – grand LP formulation is unsolved

[f] Insufficient memory available to solve problem in Grand LP formulation once copied to CPLEX

and 700,000 constraints (exact dimensions and density information not provided) and is described in Birge, et al. [14, page 346, problem SCAGR(P5S16)]. That problem was solved, however, with a parallel implementation of a nested L-Shaped decomposition algorithm on a network of eight RS/6000[6] workstations.

6.4.2 Two-Stage Decomposition

Solution CPU times as a function of relative tolerance for selected two-stage decomposition strategies applied to the 11 smallest multiple period problems are shown in Table 6.12. Times for two or more strategies applied to the same problem are shown for some problems in order to demonstrate the variations that exist between these strategies. Dantzig-Wolfe decomposition is more efficient than L-Shaped decomposition on all problems except **P2-Large**. The greatest relative time difference (approximately 93 percent) between the two methods occurs for problem **P3-Large**. Graphs of the CPU time as a function of relative tolerance for the decomposition of these two extreme problems are shown in Figure 6.4. A similar graph for each of the 11 smallest multiple period problems, including full page reproductions of Figures 6.4(a) and 6.4(b), is at Appendix G. The best DWD strategy is 20.77 percent faster than the best LSD strategy when averaged over all 11 problems and 15.46 percent faster on average when the two extreme problems are omitted. Reasons for the superior performance of DWD over LSD are examined below.

---

[6] RS/6000 is a registered trademark of International Business Machines Corporation, Armonk, NY

Table 6.12: Two-Stage Decomposition Solution CPU Times (Minutes)

Problem Label	Solution Strategy Triplet[a]	Relative Tolerance							
		Dantzig-Wolfe				L-Shaped			
		10^{-2}	10^{-4}	10^{-6}	Exact[b]	10^{-2}	10^{-4}	10^{-6}	Exact
P2-Small	(7.1.2)	0.4822	1.786	1.917	**2.042**[c]	0.5124	1.576	2.527	2.527
P2-Medium	(7.1.2)	6.235	21.90	39.14	**43.74**	6.474	24.29	46.14	52.65
P2-Large	(7.1.4)[d]	97.74	313.3	606.5	927.7	98.46	319.9	620.2	**894.7**[e]
P3-Small	(7.1.2)	9.506	31.84	46.90	46.90	9.613	32.18	49.75	49.75
	(7.2.2)	5.985	8.113	10.01	**10.29**	6.115	9.249	10.84	11.39
P3-Medium	(7.1.2)	205.0	638.1	910.8	910.8	204.8	637.6	1020	1130
	(7.2.2)	32.70	56.87	70.10	**73.96**	44.07	84.37	104.9	110.7
P3-Large	(7.2.2)	133.4	262.5	330.1	**366.9**	186.0	436.2	676.0	707.7
P4-Small	(7.1.2)	41.52	202.6	318.0	318.0	41.06	223.9	296.0	296.0
	(7.2.2)	8.063	18.47	28.75	**30.79**	8.195	18.78	29.10	31.16
	(7.3.2)	32.90	41.53	47.84	51.51	33.80	50.35	55.97	59.99
P4-Medium	(7.2.2)	107.1	255.7	434.0	**492.8**	108.0	286.8	463.4	522.1
	(7.3.2)	360.0	456.3	502.2	530.7	673.0	1,014	1,083	1,096
P4-Large	(7.2.2)	1,004	2,382	4,031	**4,854**	1,007	2,401	4,056	5,156
P5-Small	(7.2.2)	17.46	56.82	89.68	102.9	17.46	50.20	89.70	116.0
	(7.3.2)	11.43	23.65	45.89	**58.34**	13.57	29.31	44.97	63.39
	(7.4.2)	96.54	121.3	130.9	133.0	88.53	121.3	130.2	132.9
P5-Medium	(7.3.2)	138.2	276.7	483.2	**608.6**	183.2	354.9	562.5	685.6
P5-Large	Cannot be solved in reasonable amount of time with two-stage decomposition								

[a] See Section 6.2.4 and Figure 6.1 on page 239

[b] $5x10^{-8}$ is considered exact

[c] Bold face time is the fastest exact time for the individual problem

[d] Only problem for which formulation-solver (DF-DS) is best, rest use (PF-DS) – see Table 6.7

[e] Only problem for which LSD is fastest to exact – DWD is fastest for the rest

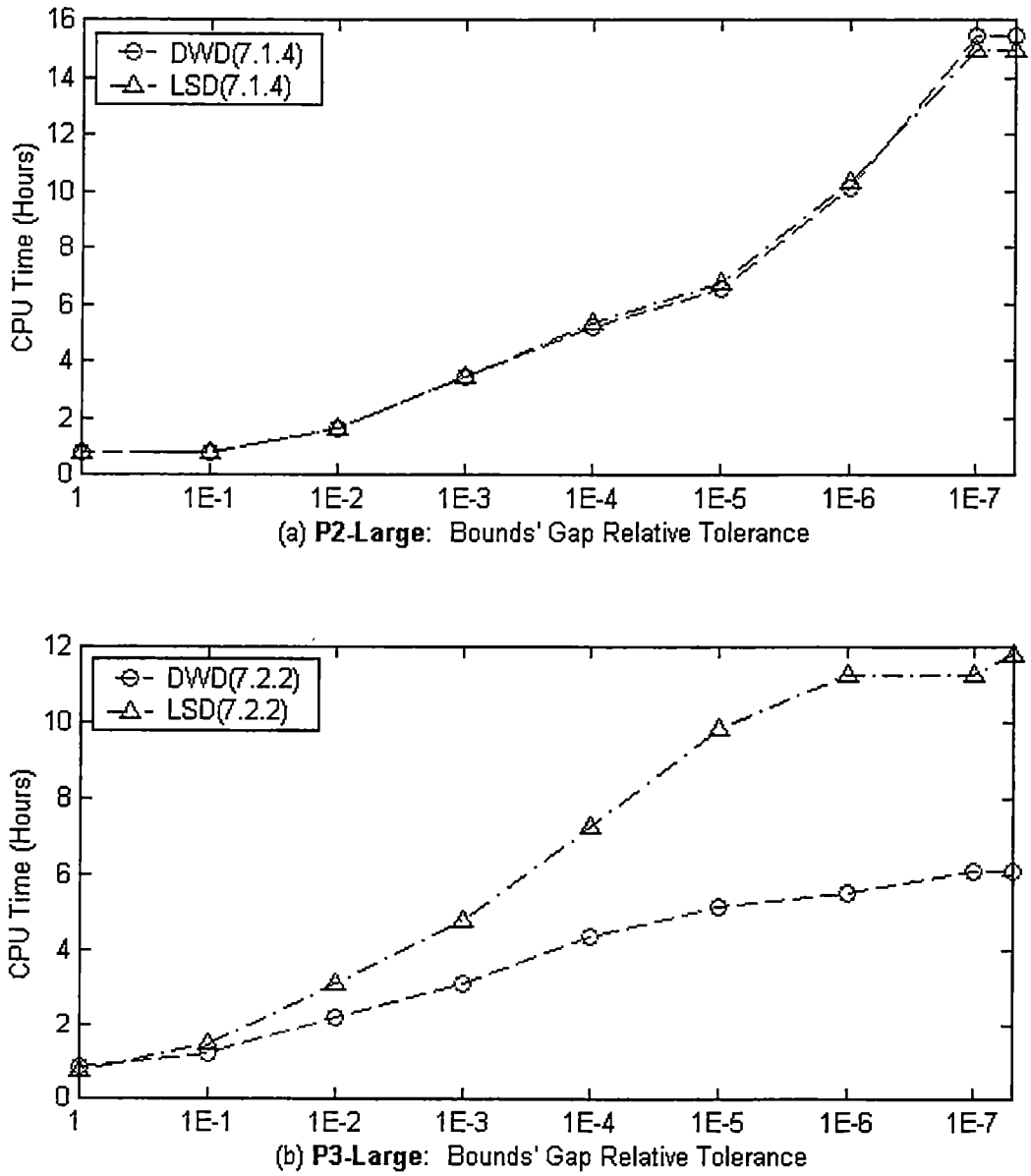


Figure 6.4: Extreme Two-Stage Decomposition Solution CPU Times

The work history of the different strategies expressed in terms of the number of iterations and the number of added cuts/activities is given in Table 6.13. This history does not explain why DWD generally outperforms LSD. For example, by comparing the complementary best strategies:

- five problems (the first four in Table 6.13 and **P3-Large**) have more LSD cuts than DWD activities added during an average iteration,
- one problem (**P3-Medium**) has the same average number of each, and
- the remaining five have fewer LSD cuts than DWD activities added on average.

Similar results hold for the other statistics listed in Table 6.13. An explanation for the performance difference between the two methods can be found, however, by a more detailed examination of the iterations data.

Two major types of iterations exist when decomposition is performed using major strategy 7:

complete – all subproblems, including grand LP nodal subproblems, are solved and both bounds can be updated, and

partial – only the simple deviation and slippage subproblems (i.e., no nodal subproblems) are solved and only the upper bound can be updated.

A complete iteration is performed as the first iteration of every decomposition procedure in order to obtain valid upper and lower bounds. Execution of complete iterations after the first is controlled by an options value provided to the *MIMPSLP* code library. This option

Table 6.13: Two-Stage Decomposition Cuts/Activities Statistics

Problem Label	Solution Strategy Triplet[a]	Dantzig-Wolfe				L-Shaped			
		Total Iters[b]	Added Activities			Total Iters	Added Cuts		
			Total[c]	Max[d]	Avg[e]		Total	Max	Avg
P2-Small	**(7.1.2)**[f]	180	1,881	102	10.45	184	1,992	102	10.83
P2-Medium	**(7.1.2)**	287	8,058	502	28.08	288	9,631	502	33.44
P2-Large	**(7.1.4)**	278	15,449	1,002	55.58	276	15,513	1,002	56.21
P3-Small	(7.1.2)	99	1,263	55	12.76	107	1,314	58	12.28
	(7.2.2)	70	23,307	1,214	332.9	69	23,570	1,173	341.6
P3-Medium	(7.1.2)	124	1,674	84	13.50	130	1,845	83	14.19
	(7.2.2)	107	60,951	4,239	569.6	109	62,083	4,239	569.6
P3-Large	**(7.2.2)**	149	123,024	10,298	825.7	147	123,783	10,298	842.1
P4-Small	(7.1.2)	65	651	27	10.01	65	636	27	9.785
	(7.2.2)	64	12,490	575	195.2	67	12,421	575	185.4
	(7.3.2)	61	98,778	6,525	1,619	59	98,512	6,521	1,670
P4-Medium	**(7.2.2)**	79	23,720	1,120	300.3	80	23,381	1,120	292.3
	(7.3.2)	73	307,900	22,709	4,218	72	307,855	22,709	4,277
P4-Large	**(7.2.2)**	94	68,145	4,240	724.9	95	68,190	4,240	717.8
P5-Small	(7.2.2)	65	8,002	363	123.1	69	8,019	357	116.2
	(7.3.2)	92	42,438	2,620	461.3	93	42,494	2,627	456.9
	(7.4.2)	60	166,470	14,029	2,775	62	166,322	14,024	2,683
P5-Medium	**(7.3.2)**	78	131,687	6,532	1,688	78	131,258	6,489	1,683
P5-Large	Not solved with two-stage decomposition								

[a] See Section 6.2.4 and Figure 6.1 on page 239

[b] Total number of iterations for relative tolerance of $5x10^{-8}$

[c] Total number of cuts/activities added over all iterations

[d] Maximum number added during any iteration (minimum varies from 1 to 16)

[e] Average number of cuts/activities added over all iterations

[f] Boldface type indicates the fastest strategy for the individual problem

value indicates that a complete iteration is to be performed at every i^{th} iteration where $i \geq 1$. Complete iterations are also automatically executed whenever the upper bound shows no improvement from one iteration to the next. The best results for model **MIMPSLP** problems are obtained with i set to a very large number so that complete iterations (after the first) are only performed after an iteration at which the upper bound fails to decrease in value. Partial iterations occur between complete iterations and serve to improve the upper bound in a greedy fashion since these iterations are significantly faster than complete iterations.

The numbers of complete and partial iterations, with the average time for each type, required for two-stage decomposition of the 11 smallest multiple period problems are shown in Table 6.14. Note that the number of partial iterations significantly exceeds the number of complete iterations in each case. In addition, the LSD procedure has a substantially greater average partial iteration time than the complementary DWD procedure for each of the 11 fastest strategy triplets. The average best DWD partial iteration time is 131.6 percent faster than the average best LSD partial iteration time ranging from a minimum of 18.7 percent (**P3-Small**) to a maximum of 240.6 percent (**P4-Large**) relative difference in favor of DWD. Times for complete iterations compare much more equably because these iterations are dominated by the time required to solve the grand LP nodal subproblems and the two decomposition methods use the same set of solution subroutines for these subproblems.

Table 6.14: Two-Stage Decomposition Average Iteration CPU Times (Seconds)

Problem[a] Label	Solution Strategy Triplet[b]	Dantzig-Wolfe			L-Shaped		
		Partial Time (#)[c]	Complete Time (#)[d]	All Time[e]	Partial Time (#)	Complete Time (#)	All Time
P2-Small	**(7.1.2)**[f]	0.0335 (164)	7.3149 (16)	0.6808	0.1022 (166)	7.4824 (18)	0.8242
P2-Medium	**(7.1.2)**	0.4829 (260)	92.55 (27)	9.144	1.4615 (259)	95.87 (29)	10.97
P2-Large	**(7.1.4)**	1.233 (243)	1,581 (35)	200.2	3.437 (243)	1,601 (33)	194.5
P3-Small	(7.1.2)	0.0176 (80)	148.0 (19)	28.42	0.0452 (87)	149.1 (20)	27.90
	(7.2.2)	6.947 (56)	16.32 (14)	8.821	8.243 (55)	16.44 (14)	9.907
P3-Medium	(7.1.2)	0.307 (107)	3,214 (17)	440.7	0.0919 (109)	3,227 (21)	521.3
	(7.2.2)	28.75 (93)	126.0 (14)	41.15	52.89 (94)	111.5 (15)	60.95
P3-Large	**(7.2.2)**	69.26 (136)	968.7 (13)	147.7	211.5 (132)	969.2 (15)	288.8
P4-Small	(7.1.2)	0.0065 (51)	1,362 (14)	293.5	0.0123 (52)	1,366 (13)	273.3
	(7.2.2)	0.5054 (49)	121.5 (15)	28.86	0.8360 (52)	121.7 (15)	27.90
	(7.3.2)	50.04 (45)	52.42 (16)	50.66	65.83 (42)	49.09 (17)	61.00
P4-Medium	**(7.2.2)**	1.379 (62)	1,734 (17)	374.3	2.646 (62)	1,731 (18)	391.6
	(7.3.2)	469.2 (57)	318.9 (16)	436.2	1,090 (57)	242.3 (15)	913.2
P4-Large	**(7.2.2)**	7.846 (76)	16,146 (18)	3,098	26.72 (76)	16,176 (19)	3,256
P5-Small	(7.2.2)	0.2510 (49)	385.0 (16)	94.96	0.3652 (51)	385.7 (18)	100.9
	(7.3.2)	7.930 (63)	103.5 (29)	38.05	11.42 (62)	99.86 (31)	40.90
	(7.4.2)	156.2 (45)	63.48 (15)	133.0	153.3 (46)	57.60 (16)	128.6
P5-Medium	**(7.3.2)**	72.91 (56)	1,474 (22)	468.1	154.2 (56)	1,477 (22)	527.4
P5-Large	Not solved with two-stage decomposition						

[a] All data for decomposition termination at relative tolerance of $5x10^{-8}$

[b] See Section 6.2.4 and Figure 6.1 on page 239

[c] Average time in seconds for (number of) partial iterations (nodal subproblems not solved)

[d] Average time in seconds for (number of) complete iterations (all subproblems solved)

[e] Average time in seconds for all iterations

[f] Boldface type indicates the fastest strategy for the individual problem

A second contributing factor to the performance difference between DWD and LSD is the time required by each of the two decomposition stages. Time distribution data for the stage one and stage two procedures of each iteration type are given in Table 6.15. L-Shaped decomposition spends a significantly greater percentage of time than Dantzig-Wolfe decomposition in stage one during the partial iterations of each of the fastest strategy triplets. Dantzig-Wolfe decomposition, on the other hand, is in stage two for both iteration types for a larger percentage of time than L-Shaped decomposition. Time spent in stage one modifying and solving the RMP is critical to the performance variation between two complementary decomposition procedures (same strategy triplet) because DWD and LSD share the same set of subproblems in stage two.

The superior performance of two-stage DWD of model **MIMPSLP** problems relative to two-stage LSD can then be mostly attributed to CPLEX. Adding variables to an existing problem of this class and resolving with an advanced basis is evidently a more efficient procedure in CPLEX than resolving the dual of that problem with a warm start after the addition of constraints.

6.4.3 Nested Decomposition

Nested decomposition was applied to the last 9 problem instances (those with three or more periods) defined in Section 6.1.2 and listed in Table 6.3 on page 231. The nested decomposition strategies used on these problems are described in Sections 4.4.4 and E.1.5. These strategies are based upon the block-separability property of model **MIMPSLP** prob-

Table 6.15: Time Percentages Devoted to Stages of Two-Stage Decomposition

Problem Label	Solution Strategy Triplet[c]	Dantzig-Wolfe				L-Shaped			
		Partial[a]		Complete[b]		Partial		Complete	
		RMP[d]	SUB[e]	RMP	SUB	RMP	SUB	RMP	SUB
P2-Small	**(7.1.2)**[f]	4.291	0.196	0.548	94.966	11.041	0.145	2.153	86.660
P2-Medium	**(7.1.2)**	4.725	0.059	0.653	94.563	11.931	0.053	3.062	84.954
P2-Large	**(7.1.4)**	0.533	0.005	0.251	99.210	1.550	0.005	0.864	97.580
P3-Small	(7.1.2)	0.484	0.002	0.034	99.916	0.129	0.003	0.095	99.773
	(7.2.2)	62.838	0.169	4.519	32.475	66.181	0.144	4.401	29.274
P3-Medium	(7.1.2)	0.006	0.000	0.002	99.992	0.015	0.000	0.008	99.977
	(7.2.2)	60.125	0.120	8.291	31.464	74.755	0.081	2.907	22.257
P3-Large	**(7.2.2)**	42.708	0.085	4.738	52.469	65.719	0.041	3.033	31.207
P4-Small	(7.1.2)	0.002	0.000	0.001	99.997	0.003	0.000	0.002	99.994
	(7.2.2)	1.318	0.228	0.507	98.152	2.302	0.023	0.409	97.266
	(7.3.2)	72.697	0.162	4.724	22.417	76.683	0.130	2.677	20.509
P4-Medium	**(7.2.2)**	0.286	0.003	0.094	99.617	0.520	0.003	0.197	99.279
	(7.3.2)	83.916	0.062	5.237	10.786	94.439	0.033	0.862	4.666
P4-Large	**(7.2.2)**	0.203	0.002	0.082	99.713	0.655	0.001	0.091	99.252
P5-Small	(7.2.2)	0.196	0.004	0.089	99.712	0.264	0.003	0.101	99.631
	(7.3.2)	14.214	0.059	3.057	82.670	18.558	0.055	1.348	80.039
	(7.4.2)	87.966	0.104	3.278	8.653	88.333	0.107	2.385	9.174
P5-Medium	**(7.3.2)**	11.164	0.017	1.765	87.054	20.980	0.016	1.664	77.340
P5-Large	Not solved with two-stage decomposition								

[a] Percentage of time in iterations during which nodal subproblems are not solved

[b] Percentage of time in iterations during which all subproblems are solved

[c] See Section 6.2.4 and Figure 6.1 on page 239

[d] Percentage of time spent in stage one

[e] Percentage of time spent in stage two

[f] Boldface type indicates the fastest strategy for the individual problem

lems. Sample procedures for three- and four-period problems are respectively illustrated by Figure E.1 on page 382 and Figure 4.4 on page 151. Only nested Dantzig-Wolfe decomposition algorithms were employed due to the generally superior performance of DWD versus LSD of this class of problems.

Solution CPU times as a function of relative tolerance for selected nested decomposition strategies applied to the subset of multiple period problems are shown in the middle portion of Table 6.16. All second stage nodal subproblems were solved to a relative tolerance equal to one-tenth of the specified termination relative tolerance for the overall problem. Nested decomposition of nodal subproblems for problem **P5-Large** was then terminated at a relative tolerance of 10^{-3}. Decomposition of all other nodal subproblems terminated at a relative tolerance of 5×10^{-9}. The solution CPU times for the fastest two-stage decomposition strategy for each problem instance are given in the last four columns of Table 6.16.

Two-stage DWD is significantly more efficient than nested DWD at each relative tolerance level for all problems except the two largest, **P4-Large** and **P5-Large**. Problem **P5-Large** is so large that the decision tree must be stored in a file and two-stage DWD cannot be applied with a minor strategy (cutoff period $C_R = \bar{t}$ for the first stage RMP) of 3 or greater under major strategy 7. Sufficient computer memory was not available to store the initial RMP and ancillary data for two-stage decomposition under major strategy 7 with $\bar{t} \geq 3$. Smaller major strategy levels are also incompatible with this problem since $\bar{t}$ must be 4 or 5 at those levels. Nested decomposition of problem **P5-Large** is faster than two-stage

Table 6.16: Nested Dantzig-Wolfe Decomposition Solution CPU Times (Minutes)

Problem Label	Solution Strategy Triplet[a]	Relative Tolerance							
		Nested Dantzig-Wolfe				Best Two-Stage			
		10^{-2}	10^{-4}	10^{-6}	Exact[b]	10^{-2}	10^{-4}	10^{-6}	Exact
P3-Small	(7.1.6)	10.14	37.59	47.63	50.15	5.985	8.113	10.01	10.29[c]
P3-Medium	(7.1.6)	75.55	282.5	392.6	392.6	32.70	56.87	70.10	73.96[d]
P3-Large	(7.1.6)	687.9	2,420	3,114	3,457	133.4	262.5	330.1	366.9[e]
P4-Small	(7.1.6)	43.52	340.5	385.2	407.7	8.063	18.47	28.75	30.79[f]
	(7.1.10)	26.48	168.8	266.9	266.9				
	(7.2.6)	24.17	55.47	86.97	99.56				
P4-Medium	(7.1.10)	265.4	998.2	1,449	1,539	107.1	255.7	434.0	492.8[g]
	(7.2.6)	130.3	333.2	569.2	636.7				
P4-Large	(7.2.6)	934.2	2,405	3,859	4,591	1,004	2,382	4,031	4,854[h]
P5-Small	(7.1.6)	65.83	613.4	883.4	883.4	11.43	23.65	45.89	58.34[i]
	(7.1.10)	66.61	584.5	960.9	960.9				
	(7.2.6)	56.48	180.5	323.7	323.7				
	(7.2.10)	33.17	103.7	173.6	196.5				
	(7.3.6)	32.14	81.98	151.1	185.5				
P5-Medium	(7.2.10)	687.5	1,525	2,526	3,362	138.2	276.7	483.2	608.6[j]
	(7.3.6)	338.5	775.9	1,348	1,629				
P5-Large	(7.2.10)	4,086	Stopped at $\varepsilon = 10^{-2}$			NA[k]			

[a] See Section 6.2.4 and Figure 6.1 on page 239

[b] $5x10^{-8}$ is considered exact

[c] Best two-stage exact time for **P3-Small** uses DWD(7.2.2)

[d] Best two-stage exact time for **P3-Medium** uses DWD(7.2.2)

[e] Best two-stage exact time for **P3-Large** uses DWD(7.2.2)

[f] Best two-stage exact time for **P4-Small** uses DWD(7.2.2)

[g] Best two-stage exact time for **P4-Medium** uses DWD(7.2.2)

[h] Best two-stage exact time for **P4-Large** uses DWD(7.2.2)

[i] Best two-stage exact time for **P5-Small** uses DWD(7.3.2)

[j] Best two-stage exact time for **P5-Medium** uses DWD(7.3.2)

[k] Not available – **P5-Large** was not solved with two-stage decomposition

decomposition when the first stage cutoff period is 2 or less because decomposition of the resulting 3- ($\bar{t} = 2$) or 4-period ($\bar{t} = 1$) nodal subproblems is more efficient than a grand LP approach. Even nested decomposition of this problem was limited to a termination relative tolerance of 10^{-2} since that procedure required over 68 hours (see the last row of Table 6.16). The size of nodal subproblems generally tends to increase with the size of the overall problem and, at some point, it is faster to solve these subproblems with decomposition than with a grand LP formulation. Therefore, nested decomposition becomes more efficient than two-stage decomposition as the size of a problem increases. Data in Table 6.16, in particular for the three 4-period problems, support the latter observation.

6.4.4 General Comments

Decomposition of stochastic linear programs in particular, and large scale linear programs in general, is clearly an attractive alternative to grand LP solution procedures. Available computer resources will limit the size of any LP that can be solved in a grand LP formulation so that some form of decomposition will often be required. Decomposition may also be significantly more efficient than a grand LP approach even for relatively small problems. The option to terminate a solution procedure at some specified bounds' gap that is less than what may be considered 'exact' is also a beneficial property of decomposition algorithms. Solution times in Table 6.12 on page 255 demonstrate, for instance, that decomposition of the 11 smallest multiple period problems can, on average, be terminated at

a relative tolerance of 10^{-2} in less than one third (0.27) of the time required for an exact solution.

Dantzig-Wolfe decomposition proved to be more efficient than L-Shaped decomposition for model **MIMPSLP** problems over planning horizons of one to five periods. Employment of CPLEX software may be a critical factor in this performance variation. The difference in efficiencies may also not be present, or even reversed, in other problem classes. It is important to realize, however, that such differences may exist between these two decomposition methods. Knowledge of the relative efficiencies of Dantzig-Wolfe and L-Shaped decomposition for a given class of problems may be necessary to the design of the most effective solution procedures.

Two-stage decomposition is generally more efficient than nested decomposition when sufficient resources are available to execute both procedures. Many stochastic linear programs possess the block-separability property and can be structured for two-stage decomposition. Nested decomposition, however, may be required or may be more efficient than two-stage decomposition for very large problems. Block-separability, if present, can also be incorporated to optimize the design of nested decomposition algorithms.

6.5 Myopic Decomposition Results

Myopic decomposition as described in Section 4.5 starting on page 152 was applied to the 12 multiple period problem instances defined in Section 6.1.2 and listed in Table 6.3

on page 231. Both the primals lead and duals lead algorithms detailed in Appendix B were applied to each problem instance. Only the results for the primals lead algorithm are discussed below since this version significantly outperformed the duals lead version in all cases.

Myopic decomposition terminated for each problem instance after the completion of a full cycle with no improvement in the bounds' gap from the previous cycle. Performance data at termination and after the first complete cycle are shown in Table 6.17. The last three columns of this table list data for the fastest two-stage decomposition method and strategy triplet for each problem.

Data in Table 6.17 indicates that there was no significant improvement in the bounds' gap of any problem after the first cycle. In fact, there was no improvement after the first cycle for 6 of the 12 problems (termination after 2 cycles). Note, however, that the first cycle data for myopic decomposition compares very favorably with the data for the fastest two-stage decomposition procedures. Myopic decomposition achieved the first cycle bounds' gap significantly faster than achieved by the fastest two-stage decomposition method for all problems except **P2-Large** for which the two times are nearly equal. Myopic decomposition required, on average, 27 percent of the time required by two-stage decomposition to obtain the first cycle bounds' gap.

Myopic decomposition lower and upper bounds at termination and after the first complete are shown in Table 6.18 along with the fractional gap between the lower bound and the exact objective function value obtained with Dantzig-Wolfe and L-Shaped decomposition.

Table 6.17: Myopic Decomposition Solution CPU Times (Minutes)

Problem Label	Myopic Decomposition Data					Two-Stage Data[a]		Time Ratio[e]
	Termination Data			1st Cycle Data		Method (Strategy)[c]	Nearest Time[d]	
	% Gap	Time	Cycles[b]	% Gap	Time			
P2-Small	1.129	0.3009	2	1.129	0.1524	DWD(7.1.2)	0.4822	0.3161
P2-Medium	1.598	5.083	3	1.828	1.703	DWD(7.1.2)	4.621	0.3685
P2-Large	2.158	155.7	3	2.210	51.91	LSD(7.1.4)	51.63	1.005
P3-Small	2.496	0.9470	2	2.496	0.4635	DWD(7.2.2)	5.367	0.08636
P3-Medium	2.448	5.242	2	2.448	2.603	DWD(7.2.2)	32.70	0.07960
P3-Large	2.726	34.72	2	2.726	17.39	DWD(7.2.2)	107.0	0.1625
P4-Small	3.266	4.160	2	3.266	2.067	DWD(7.2.2)	5.923	0.3490
P4-Medium	3.610	16.75	2	3.610	8.333	DWD(7.2.2)	77.58	0.1074
P4-Large	4.026	136.3	3	4.244	45.14	DWD(7.2.2)	728.5	0.06196
P5-Small	4.865	7.967	3	4.944	2.647	DWD(7.3.2)	7.008	0.3777
P5-Medium	4.903	85.86	4	5.559	21.44	DWD(7.3.2)	79.39	0.2701
P5-Large	5.582	341.9	3	5.823	113.3	DWD(7.2.10)	1,862	0.06085

[a] Data for the fastest two-stage Dantzig-Wolfe or L-Shaped decomposition procedure

[b] Index of cycle at which there was no improvement in bounds gap from previous cycle

[c] Decomposition method acronym and strategy triplet

[d] Time required for two-stage method to achieve the 1 cycle myopic bounds gap or better

[e] Ratio of the 1 cycle myopic time to the two-stage time

Table 6.18: Myopic Decomposition Objective Value Comparisons

Problem Label	Exact Value Obj[a]	Myopic Decomposition				Fractional Gap To Lower Bound	
		1st Cycle Bounds		Termination Bounds			
		Lower MLB_1	Upper MUB_1	Lower MLB_F	Upper MUB_F	1st[b]	Term[c]
P2-Small	2,822,734	2,804,928	2,836,588	2,804,928	2,836,588	0.5624	0.5624
P2-Medium	2,596,361	2,568,460	2,615,418	2,574,269	2,615,418	0.5942	0.5942
P2-Large	2,472,811	2,445,419	2,499,485	2,446,677	2,499,485	0.5066	0.5066
P3-Small	3,022,102	2,992,621	3,067,329	2,992,621	3,067,329	0.3946	0.3946
P3-Medium	2,496,515	2,476,574	2,537,189	2,476,574	2,537,189	0.3290	0.3290
P3-Large	2,152,676	2,127,553	2,185,548	2,127,553	2,185,548	0.4332	0.4332
P4-Small	4,326,598	4,272,048	4,411,582	4,272,048	4,411,582	0.3909	0.3909
P4-Medium	3,972,196	3,916,018	4,057,380	3,916,018	4,057,380	0.3974	0.3974
P4-Large	3,692,139	3,624,752	3,778,584	3,632,340	3,778,584	0.4381	0.4089
P5-Small	3,927,228	3,843,899	4,033,935	3,846,775	4,033,935	0.4385	0.4299
P5-Medium	3,696,311	3,618,894	3,820,076	3,620,336	3,797,832	0.3848	0.4291
P5-Large	3,588,547[d]	3,520,603	3,725,592	3,525,459	3,722,253	0.3315	0.3206

[a] Exact DWD/LSD objective function values (5×10^{-8} relative tolerance) except for problem **P5-Large**

[b] $\frac{Obj - MLB_1}{MUB_1 - MLB_1}$

[c] $\frac{Obj - MLB_F}{MUB_F - MLB_F}$

[d] Lower Bound at 1 percent relative gap with DWD(7.2.10)

The exact objective function value is nearer the lower bound (fractional gap is less than 0.5) for all problems except the three two-period problems. Future research should then be directed toward improving the tail (dual) half-cycle results in order to obtain smaller upper bounds at each cycle.

The biggest drawback to the current myopic decomposition algorithms is their inability to guarantee a specified relative tolerance. They can, however, be very efficient relative to Dantzig-Wolfe and L-Shaped decomposition in obtaining fairly good solutions.

6.6 Sample Effectiveness Measures

The Mahalanobis squared distance based measures developed in Section 5.6.4 starting on page 216 may be used to supplement traditional methods of judging the effectiveness of a random sample in approximating a known distribution. Mahalanobis squared distance values and sample effectiveness measures for six model **MIMPSLP** problem instances are described below. Only problems with a single period are evaluated since the scenario generator library (see Section 5.6 starting on page 207) employs a multivariate normal distribution to generate random outcomes in each period of the planning horizon. Therefore, no additional insight into the value of the proposed measures is gained by examining model **MIMPSLP** problems with more than one period.

Three random seed values were used to generate outcomes for each of the six problems which, otherwise, differ only in the number of outcomes in the random sample. The

resulting expected population and average sample Mahalanobis squared distance values defined by equations [5.64] through [5.75] starting on page 220 are listed in Table 6.19. These values were then used to determine the six sample effectiveness measures defined by equation [5.76] on page 224 and the results are given in Table 6.20.

Several interesting conjectures may be formulated based upon the data in Table 6.20. Measures based upon comparing the population expected return with the sample average return ($E_2^{(1)}$) and the expected population net return with the average sample net return ($E_5^{(1)}$) do not appear to be useful since they are uniformly relatively large and exhibit substantial variations. Each of the remaining four measures are based upon comparing sample returns and net returns vectors with the corresponding population expected and sample average returns and net returns vectors. These measures tend to behave as expected since they generally decrease in value with increasing sample size. Measures based upon Mahalanobis squared distances between sample net returns vectors and the population expected net returns vector ($E_4^{(1)}$) and the sample average net returns vector ($E_6^{(1)}$) are considered the more important pair. The latter two measures characterize the results of the model's investments decisions whereas measures $E_1^{(1)}$ and $E_3^{(1)}$ provide information only for the generated sample.

Values in Table 6.20 for measures $E_4^{(1)}$ and $E_6^{(1)}$ indicate that a sample size of from 20 to 50 outcomes is probably too small to provide investment decisions that are robust for different seed values. Values of these measures for samples of 100 to 1,000 outcomes, on the other hand, vary little across both the seed values for the same sample size and the

Table 6.19: Mahalanobis Squared Distance Values for Single Period Problems

Outcomes	Random	Returns[a]			Net Returns[b]		
L	Seed	$\mathcal{M}_1^{(1)}$	$\mathcal{M}_2^{(1)}$	$\mathcal{M}_3^{(1)}$	$\mathcal{M}_4^{(1)}$	$\mathcal{M}_5^{(1)}$	$\mathcal{M}_6^{(1)}$
20	100	68[c]	3.40000	64.60000	26	1.30000	24.70000
		68.66788[d]	2.26265	66.40523	27.16844	1.04168	27.77348
	1,000	68	3.40000	64.60000	26	1.30000	24.70000
		67.30189	2.84333	64.45857	25.84362	1.05348	26.20875
	10,000	68	3.40000	64.60000	24	1.20000	22.80000
		67.00530	3.94039	63.06491	22.93230	1.31420	21.52928
50	100	68	1.36000	66.64000	23	0.46000	22.54000
		67.59106	0.94155	66.64951	23.08775	0.25904	20.75519
	1,000	68	1.36000	66.64000	23	0.46000	22.54000
		68.11555	1.46123	66.65432	24.28390	0.59530	24.74113
	10,000	68	1.36000	66.64000	27	0.54000	26.46000
		68.18476	1.49929	66.68547	27.34499	0.59303	24.07765
100	100	68	0.68000	67.32000	29	0.29000	28.71000
		68.59674	0.49566	68.10108	29.84762	0.16944	29.40369
	1,000	68	0.68000	67.32000	25	0.25000	24.75000
		68.18315	0.78266	67.40049	25.56521	0.32157	24.77922
	10,000	68	0.68000	67.32000	28	0.28000	27.72000
		67.99100	0.59751	67.39349	28.69767	0.20103	26.12339
500	100	68	0.13600	67.86400	31	0.06200	30.93800
		69.09147	0.16018	68.93129	31.86859	0.08689	31.64606
	1,000	68	0.13600	67.86400	27	0.05400	26.94600
		68.39446	0.18370	68.21076	27.46871	0.09001	27.09422
	10,000	68	0.13600	67.86400	31	0.06200	30.93800
		68.47139	0.14105	68.33034	31.29956	0.08562	30.64816
1,000	100	68	0.06800	67.93200	29	0.02900	28.97100
		68.37054	0.05637	68.31417	29.20934	0.02507	29.04360
	1,000	68	0.06800	67.93200	27	0.02700	26.97300
		68.32987	0.08138	68.24849	27.31412	0.04526	27.47360
	10,000	68	0.06800	67.93200	28	0.02800	27.97200
		68.00626	0.05797	67.94829	28.05830	0.03036	27.52237
100,000	100	68	0.00068	67.99932	28	0.00028	27.99972
		67.97271	0.00063	67.97209	27.98512	0.00032	27.98232
	1,000	68	0.00068	67.99932	28	0.00028	27.99972
		68.03666	0.00069	68.03597	27.99650	0.00023	28.00022
	10,000	68	0.00068	67.99932	28	0.00028	27.99972
		68.02099	0.00071	68.02027	27.99423	0.00040	27.99423

[a] See equations [5.64] through [5.69] starting on page 220

[b] See equations [5.70] through [5.75] starting on page 223

[c] Population expected value is in the top half of each cell

[d] Sample average value is in the bottom half of each cell

Table 6.20: Sample Effectiveness Measures for Single Period Problems

Outcomes	**Random**	**Returns**[a]			**Net Returns**[b]		
L	**Seed**	$E_1^{(1)}$	$E_2^{(1)}$	$E_3^{(1)}$	$E_4^{(1)}$	$E_5^{(1)}$	$E_6^{(1)}$
20	100	0.98218	33.451	2.7945	4.4940	19.871	12.443
	1,000	1.0266	16.373	0.21893	0.60146	18.963	6.1083
	10,000	1.4628	15.894	2.3763	4.4488	9.5167	5.5733
50	100	0.60138	30.768	0.014271	0.38152	43.687	7.9184
	1,000	0.16993	7.4434	0.021489	5.5822	29.413	9.7654
	10,000	0.27171	10.242	0.068232	1.2777	9.8204	9.0036
100	100	0.87756	27.109	1.1602	2.9228	41.572	2.4162
	1,000	0.19338	15.097	0.11956	2.2608	28.628	0.11806
	10,000	0.013235	12.131	0.10917	2.4917	28.204	5.7598
500	100	1.6051	17.779	1.5727	2.8019	40.145	2.2886
	1,000	0.58009	35.074	0.51096	1.7360	66.685	0.54999
	10,000	0.69322	21.074	0.68717	0.96632	38.097	0.93684
1,000	100	0.54491	17.103	0.56258	0.72186	13.552	0.25060
	1,000	0.48510	19.676	0.46589	1.1634	67.630	1.8559
	10,000	0.0092059	14.750	0.041652	0.20821	8.4286	1.6074
100,000	100	0.040132	7.3529	0.040045	0.053143	14.286	0.062144
	1,000	0.053912	1.4706	0.053898	0.012500	17.857	0.0017857
	10,000	0.030868	4.4118	0.030809	0.020607	42.857	0.019607

[a] See equation [5.76] on page 224 with $j = 1, 2, 3$

[b] See equation [5.76] on page 224 with $j = 4, 5, 6$

different sample sizes. Samples with 100,000 outcomes exhibit uniformly small $E_4^{(1)}$ and $E_6^{(1)}$ (also $E_1^{(1)}$ and $E_3^{(1)}$) values for all random seeds and should be more than adequate for the model.

The above observations can be used to supplement information gathered from a more traditional approach such as judging model robustness to different sample sizes and random seed values. For example, the net return for each random seed and sample size combination used above is listed in Table 6.21. Note that net returns for samples with 100 or more

Table 6.21: Net Returns for Single Period Problems

Outcomes	**Random Seed**		
L	**100**	**1,000**	**10,000**
20	143,739.8	126,506.0	110,304.0
50	162,230.4	161,596.1	125,322.9
100	161,194.9	168,310.9	151,691.2
500	166,963.7	169,969.2	165,991.1
1,000	170,199.4	173,320.0	168,129.2
100,000	170,683.7	171,146.0	171,328.0

outcomes vary much less than the smaller samples across seed values. In addition, the net returns for samples with from 100 to 1,000 outcomes are relatively close to the fairly uniform results for a sample with 100,000 outcomes – a maximum difference of 11.5 percent between net returns for 100 outcomes and 100,000 outcomes with a common seed value of 10,000. These observations agree those based upon the Mahalanobis squared distance sample effectiveness measures. This agreement suggests that the proposed measures could prove to be very useful analysis tools.

6.7 Model Application Results

Model **MIMPSLP** incorporates uncertainties about future securities' performances to formulate investment decisions that reflect the individual investor's aversion to risk. Model results may be used to create analysis tools to aid in the final decision making process. Two such commonly used tools are the *efficient frontier* and the *Sharpe ratio.*

Markowitz [142] introduced the efficient frontier in 1959 as a method to identify the portfolio with the highest rate of return at a given level of risk. The efficient frontier is a concave curve that is the upper boundary on the rate of return for a collection of risky investments. The Sharpe ratio is used to measure the reward to risk performance of a particular investment. First introduced by Sharpe [187, page 123] in 1966 as the *reward-to-variability* ratio, the measure has commonly become known as the Sharpe ratio (see Sharpe [188, page 688]). These two tools are illustrated in Figure 6.5.

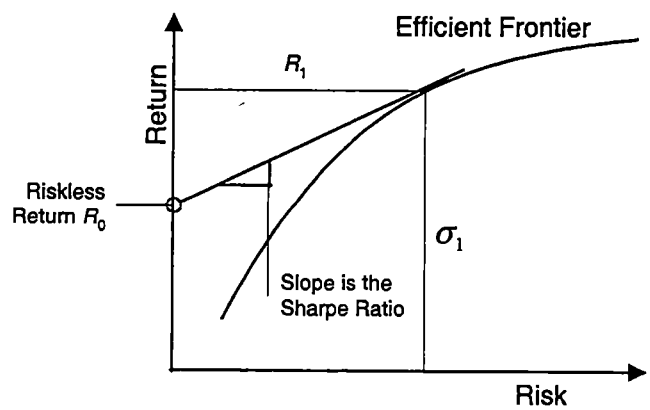


Figure 6.5: Efficient Frontier and the Sharpe Ratio

The portfolio with the maximum possible rate of return among all portfolios with a given level of risk is the one that lies on the efficient frontier boundary depicted in Figure 6.5. Sharpe ratios are commonly used to determine the investment with the maximum rate of return relative to some riskless investment. For example, given the riskless investment in Figure 6.5 with return R_0, the line through R_0 and tangent to the efficient frontier identifies the desired portfolio as the one with risk σ_1 and return R_1. The corresponding Sharpe ratio, SR, is then:

$$SR = \frac{(R_1 - R_0)}{\sigma_1}.$$

Problem **P3-Small** is used as a basis to demonstrate the coupling of these two analysis tools with results of the market investment model. Fifteen problem instances with graduated aversion to risk levels were created. The analysis in Section 6.6 above indicates that a sample of 50 first period outcomes generated for a problem **P3-Small** instance probably does not adequately represent the population. Therefore, a post solution simulation option available in the *MIMPSLP* library is utilized to estimate risk data based upon a sample of 1,000 first period outcomes for each problem instance. Risk is determined as the product of the expected value of the displacement below the mean return (expected shortfall deviation) with the fraction of the risk budget that is at risk (one minus the fraction of slack in the risk budget constraint). This risk measure then combines both of the model's risk components – the risk budget constraint and the shortfall from the mean return. The first period efficient frontier based upon expected net returns and this risk measure is illustrated in Figure 6.6. The concave shape of the efficient frontier depicted in Figure 6.6 agrees with the

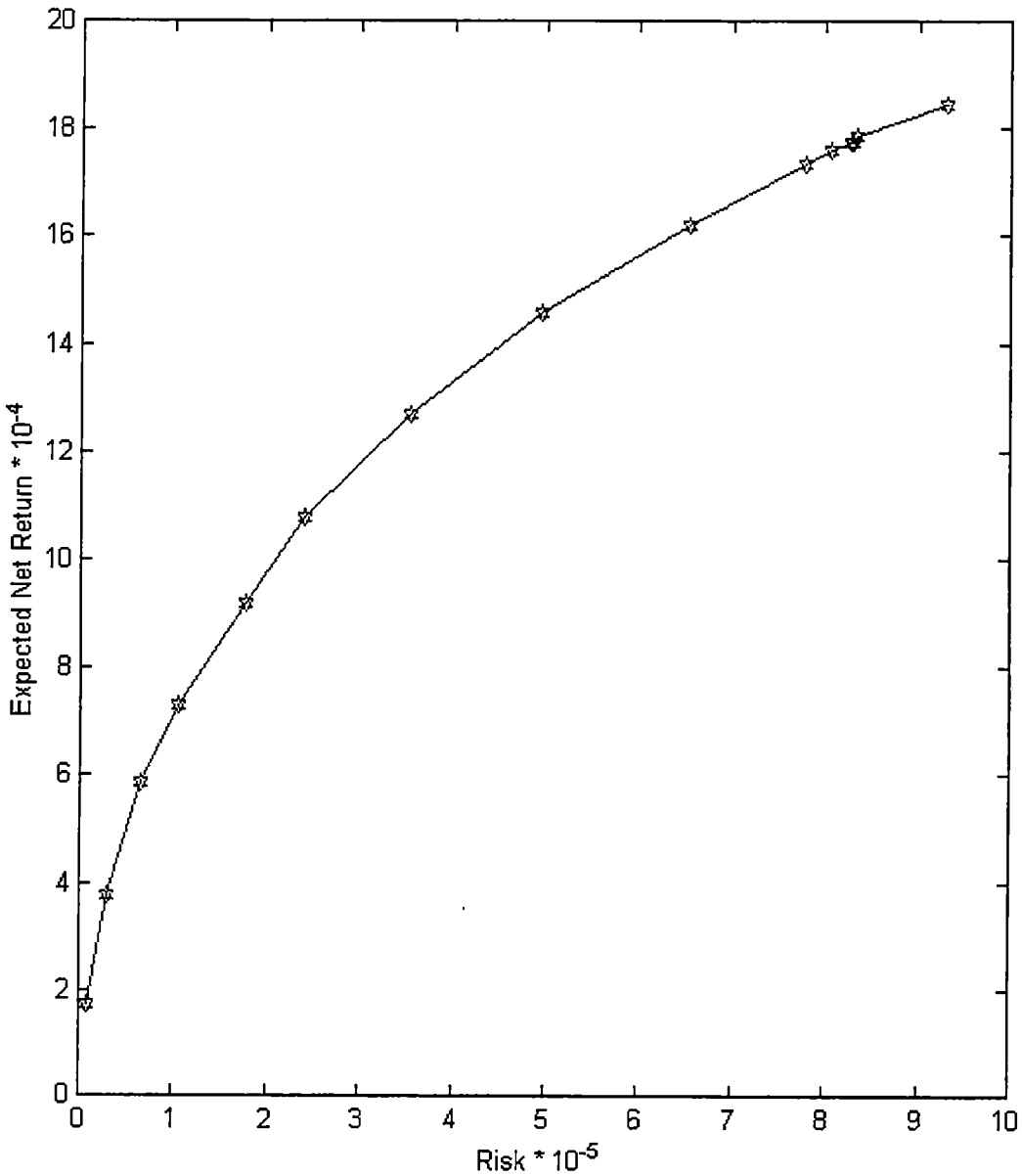


Figure 6.6: Problem **P3-Small** First Period Efficient Frontier

classical shape illustrated by Figure 6.5. Decreasing aversion to risk values (increasing risk measures) yield increasing expected net returns but at higher levels of risk.

The *MIMPSLP* code library currently evaluates Sharpe ratios based upon a riskless return rate of zero. Sharpe ratios, scaled to daily values, for the expected net returns of the 15 problem **P3-Small** instances are plotted in Figure 6.7 versus the same risk measure (scaled to daily values) used in the efficient frontier illustration at Figure 6.6. Note that the resulting curve clearly indicates that there is an optimal (relative to Sharpe ratios) level of risk. The optimal risk level from Figure 6.7 is very small indicating that a high aversion to risk is warranted. This result is expected since a riskless return rate of zero yields a steep efficient frontier tangent line with a small risk level at the point of tangency (see Figures 6.5 and 6.6).

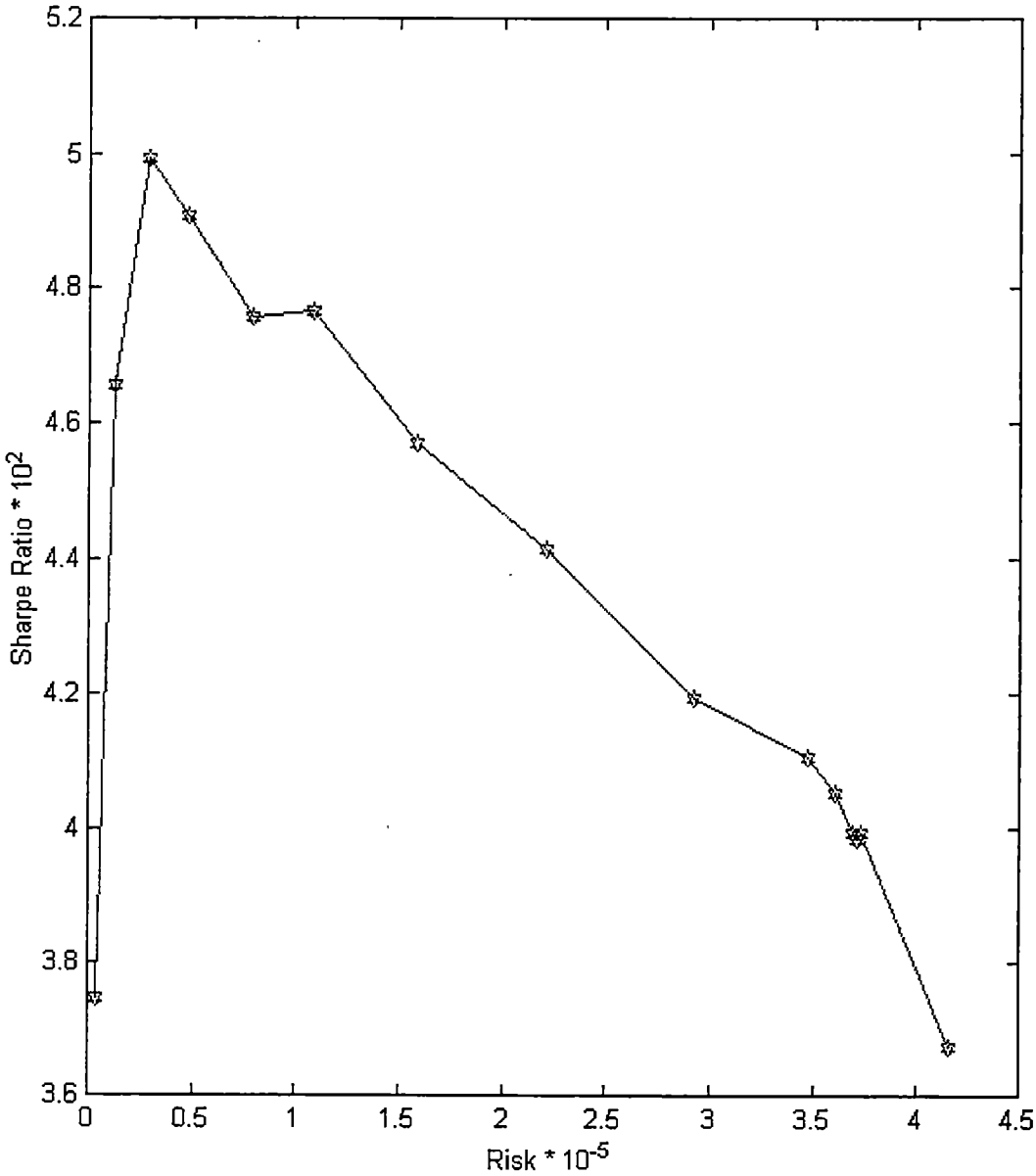


Figure 6.7: Problem **P3-Small** First Period Sharpe Ratios

Chapter 7
Summary and Conclusions

Stochastic linear programming is an effective and often used tool for incorporating knowledge of uncertainties into decision making processes. Problems based upon stochastic programming techniques are generally quite large and require sophisticated solution procedures. Developing and implementing such procedures are the primary goals of this thesis.

Effective solution algorithms require efficient structures and procedures for data storage and retrieval. Analysis tools for measuring the effectiveness of a random sample and judging a model's results are also necessary components of a comprehensive mathematical model. Efficient methods for addressing each of these areas are secondary goals.

Detailed solution algorithms for multiple period stochastic linear programs are developed based upon Dantzig-Wolfe and L-Shaped decomposition. These algorithms allow for solutions to within an arbitrary tolerance on the gap between the lower and upper bounds of a problem's objective function value. Results obtained by applying the decomposition algorithms to a multiple period market investment model are reported and analyzed. Very large market investment problems, with tens of millions of constraints and variables, were solved on a personal computer. Three, hopefully significant, contributions derived from this work are described in the following list.

Chapter 7 Summary and Conclusions

1. First known application of solution procedures based upon both Dantzig-Wolfe and L-Shaped decomposition to the same class of problems. The results indicate that one method may, in practice, be significantly more efficient than the other for certain classes of problems. This observation counters the generally accepted theory that procedures based upon the two decomposition methods should have nearly equivalent efficiencies.

2. First known computational study of the application of the block-separability property possessed by many stochastic linear programs. Results indicate that two-stage decomposition can be significantly more efficient than nested decomposition on block-separable problems with from 1 to 5 periods and the constaints and variables number on the order of millions or less.

3. First known development of nested decomposition algorithms that take advantage of the block-separability property. These algorithms allow for greater flexibility in structuring block-separable problems for decomposition than allowed for by currently documented algorithms. The resulting nested decomposition procedures are more efficient than two-stage decomposition procedures on very large problems (tens of millions of constraints and variables).

A third decomposition technique based upon a myopic view of the future is developed and applied to market investment problems. Myopic decomposition algorithms can yield very good solutions in a fraction of the time required by Dantzig-Wolfe/L-Shaped decomposition algorithms. Inability to guarantee an arbitrary solution tolerance is the major drawback to the current myopic decomposition algorithms.

Chapter 7 Summary and Conclusions

Mahalanobis squared distance based sample effectiveness measures are developed and shown to be useful supplements to traditional tools such as analysis based upon variations in the samples' sizes and generation seeds. Data storage and retrieval procedures based upon a multiple period decision tree structure are described and implemented. These techniques are very efficient and require only a fraction (negligible for large problems) of the processing time required by the selected solution procedure.

Future research is recommended in two areas. First, Dantzig-Wolfe and L-Shaped decomposition algorithms should be applied with a broad selection of simplex solvers to varied classes of problems to study the relative efficiencies of the two methods. Knowledge of these relative efficiencies could prove to be very beneficial to the design of effective solution procedures. Second, parallel computing algorithms should be developed to take advantage of block-separability when present. Extremely large multiple period problems could then be efficiently solved with two-stage decomposition methods.

REFERENCES

References

[1] Abrahamson, P. 1981. A nested decomposition approach for solving staircase-structured linear programs. In: G. B. Dantzig, M. A. H. Dempster, and M. Kallio, eds., *Large-Scale Linear Programming*, Volumn 1, 367-381.

[2] Alonso, A., L. F. Escudero, and M. T. Ortuno. 2000. A stochastic 0-1program based approach for the air traffic flow management problem. *European Journal of Operational Research* 120, 47-62.

[3] Ament, D., J. Ho, E. Loute, and M. Remmelswaal. 1981. LIFT: a nested decomposition algorithm for solving lower block triangular linear programs. In: G. B. Dantzig, M. A. H. Dempster, and M. Kallio, eds., *Large-Scale Linear Programming*, Volumn 1, 383-408.

[4] Anandalingam, G. 1987. A stochastic programming process model for investment planning. *Computers & Operations Research* 14(6), 521-536.

[5] Bahn, O., O. du Merle, J.-L. Goffin, and J.-P. Vial. 1995. A cutting plane method from analytic centers for stochastic programming. *Mathematical Programming* 69, 45-73.

[6] Bazaraa, M. S., J. J. Jarvis, and H. D. Sherali. 1990. *Linear Programming and Network Flows*. John Wiley & Sons, New York.

[7] Bazaraa, M. S., H. D. Sherali, and C. M. Shetty. 1993. *Nonlinear Programming, Theory and Algorithms*, Second Edition. John Wiley & Sons, Inc., New York.

[8] Beale, E. M. L. 1955. On minimizing a convex function subject to linear inequalities. *Journal of the Royal Statistical Society, Series B* 17, 173-184.

[9] Benders, J. F. 1962. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik* 4, 238-252.

[10] Bienstock, D. and J. F. Shapiro. 1988. Optimizing resource acquisition decisions by stochastic programming. *Management Science* 34(2), 215-229.

[11] Birge, J. R. 1985. Decomposition and partitioning methods for multistage stochastic linear programs. *Operations Research* 33(5), 989-1007

References

[12] Birge, J. R. 1985. A Dantzig-Wolfe decomposition variant equivalent to basis factorization. In: R. W. Cottle, ed., *Mathematical Programming Study 24, 43-64.*

[13] Birge, J. R., M. A. H. Dempster, H. I. Gassmann, E. A. Gunn, A. J. King, and S. W. Wallace. 1987. A standard input format for multiperiod stochastic linear programs. *Mathematical Programming Society, Committee On Algorithms Newsletter 17*, 1-19. (Paper obtained as file *smps.ps* on the World Wide Web, *http://ttg.sba.dal.ca/sba/profs/hgassmann/smps.html*)

[14] Birge, J. R., C. J. Donohue, D. F. Holmes, and O. G. Svintsitski. 1996. A parallel implementation of the nested decomposition algorithm for multistage stochastic linear programs. *Mathematical Programming* 75, 327-352.

[15] Birge, J. R. and D. F. Holmes. 1992. Efficient solution of two-stage stochastic linear programs using interior point methods. *Computational Optimization and Applications* 1, 245-276.

[16] Birge, J. R. and F. V. Louveaux. 1988. A multicut algorithm for two-stage stochastic linear programs. *European Journal of Operational Research* 34, 384-392.

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

[18] Birge, J. R. and C. H. Rosa. 1995. Modeling investment uncertainty in the costs of global CO_2 emission policy. *European Journal of Operational Research* 83, 466-488.

[19] Birge, J. R. and R. J-B. Wets. 1986. Designing approximation schemes for stochastic optimization problems, in particular for stochastic programs with recourse. In: A. Prekopa and R. J-B. Wets, eds., *Mathematical Programming Study* 27, 54-102.

[20] Birge, J. R. and R. J-B. Wets. 1987. Computing bounds for stochastic programming problems by means of a generalized moment problem. *Mathematics of Operations Research* 12(1), 149-162.

[21] Bitran, G. R. and S. Dasu. 1992. Ordering Policies in an environment of stochastic yields and substitutable demands. *Operations Research* 40(5), 999-1017.

[22] Bloom, J. A. 1983. Solving an electricity generating capacity expansion planning problem by generalized Benders' decomposition. *Operations Research* 31(1), 84-100.

[23] Bloom, J. A., M. Caramanis, and L. Charny. 1984. Long-range generation planning using generalized Benders' decomposition: implementation and experience. *Operations Research* 32(2), 290-313.

[24] Bonanno, A. S. and P. R. Griffiths. 1995. Discrimination of organic solvents using an infrared-emitting diode-based analyzer. Part I: feasibility. *Applied Spectroscopy* 49(11), 1590-1597.

[25] Borison, A. B., P. A. Morris, and S. S. Oren. 1984. A state-of-the-world decomposition approach to dynamics and uncertainty in electric utility generation expansion planning. *Operations Research* 32(5), 1052-1068.

[26] Carino, D. R., T. Kent, D. H. Myers, C. Stacy, M. Sylvanus, A. L. Turner, K. Watanabe, and W. T. Ziemba. 1994. The Russell-Yasuda Kasai Model: An asset/liability model for a Japanese insurance company using multistage stochastic programming. *Interfaces* 24(1), 29-49.

[27] Carino, D. R., D. H. Myers, and W. T. Ziemba. 1998. Concepts, technical issues, and uses of the Russell-Yasuda Kasai financial planning model. *Operations Research* 46(4), 450-461.

[28] Carino, D. R. and W. T. Ziemba. 1998. Formulation of the Russell-Yasuda Kasai financial planning model. *Operations Research* 46(4), 433-449.

[29] Charnes, A. and W. W. Cooper. 1960. Chance-constrained programming. *Management Science* 6(1), 73-79.

[30] Charnes, A., W. W. Cooper, and G. H. Symonds. 1958. Cost horizons and certainty equivalents: an approach to stochastic programming of heating oil. *Management Science* 4(3), 235-263.

[31] Chen, Z. L. and W. B. Powell. 1999. Convergent cutting-plane and partial-sampling algorithm for multistage stochastic linear programs with recourse. *Journal of Optimization Theory and Applications* 102(3), 497-524.

[32] Chun, B. J. and S. M. Robinson. 1995. Scenario analysis via bundle decomposition. *Annals of Operations Research* 56, 39-63.

[33] Chung, K. L. 1979. *Elementary Probability Theory with Stochastic Processes.* Springer-Verlag, New York.

[34] Chvatal, V. 1983. *Linear Programming.* W. H. Freeman and Company, New York.

[35] Clay, R. L. and I. E. Grossmann. 1997. A disaggregation algorithm for the optimization of stochastic planning models. *Computers & Chemical Engineering* 21(7), 751-774.

[36] Couillard, J. 1993. A decision support system for vehicle fleet planning. *Decision Support Systems* 9, 149-159.

[37] Couillard, J. and A. Martel. 1990. Vehicle fleet planning in the road transportation industry. *IEEE Transactions on Engineering Management* 37(1), 31-36.

[38] Consigli, G. and M. A. H. Dempster. 1998. Dynamic stochastic programming for asset-liability management. *Annals of Operations Research* 81, 131-161.

[39] CRC. 1996. *Standard Mathematical Tables and Formulae*, Thirtieth Edition. D. Zwillinger, editor-in-chief. CRC Press, Inc., Boca Raton.

[40] Dantzig, G. B. 1949. Programming of independent activities II: mathematical model. *Econometrica* 17(3-4), 200-211.

[41] Dantzig, G. B. 1955. Linear programming under uncertainty. *Management Science* 1(3-4), 197-206.

[42] Dantzig, G. B. 1959. On the status of multistage linear programming problems. *Management Science* 6(1), 53-72.

[43] Dantzig, G. B. 1963. *Linear Programming and Extensions.* Princeton University Press, Princeton, NJ.

[44] Dantzig, G. B. 1981. Time-staged methods in linear programming: comments and early history. In: G. B. Dantzig, M. A. H. Dempster, and M. Kallio, eds., *Large-Scale Linear Programming*, Volume 1, 3-16.

[45] Dantzig, G. B. and P. W. Glynn. 1990. Parallel processors for planning under uncertainty. *Annals of Operations Research* 22, 1-21.

[46] Dantzig, G. B. and G. Infanger. 1993. Multi-stage stochastic linear programs for portfolio optimization. *Annals of Operations Research* 45, 59-76.

[47] Dantzig, G. B. and A. Madansky. 1961. On the solution of two-stage linear programs under uncertainty. In: J. Neyman, ed., *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability,* 165-176.

[48] Dantzig, G. B. and M. N. Thapa. 1997. *Linear Programming 1: Introduction.* Springer-Verlag, New York.

[49] Dantzig, G. B. and P. Wolfe. 1960. Decomposition principle for linear programs. *Operations Research* 8, 101-111.

[50] Dantzig, G. B. and P. Wolfe. 1961. The decomposition algorithm for linear programs. *Econometrica* 29(4), 767-778.

[51] Darby-Dowman, K., S. Barker, E. Audsley, and D. Parsons. 2000. A two-stage stochastic programming with recourse model for determining robust planting plans in horticulture. *Journal of the Operational Research Society* 51, 83-89.

[52] DeGroot, M. H. 1970. *Optimal Statistical Decisions.* McGraw-Hill, Inc., New York.

[53] Dempster, M. A. H. and R. T. Thompson. 1999. EVPI-based importance sampling solution procedures for multistage stochastic linear programmes on parallel MIMD architectures. *Annals of Operations Research* 90, 161-184.

[54] Dia, X. and S. Khorram. 1998. The effects of image misregistration on the accuracy of remotely sensed change detection. *IEEE Transaction on Geoscience and Remote Sensing* 36(5), 1566-1577.

[55] Dror, M. 1993. Modeling vehicle routing with uncertain demands as a stochastic program: properties of the corresponding solution. *European Journal of Operational Research* 64, 432-441.

[56] Duffuaa, S. O. and K. S. Al-Sultan. 1999. A stochastic programming model for scheduling maintenance personnel. *Applied Mathematical Modelling* 25, 385-397.

[57] Dupacova, J. 1974. Minimax stochastic programs with nonconvex nonseparable penalty functions. *Colloquia Mathematica Societatis Janos Bolyai*, 303-316.

[58] Dupacova, J. 1987. Stochastic programming with incomplete information: a survey of results on postoptimization and sensitivity analysis. *Optimization* 18(4), 507-532.

[59] Dupacova, J., M. Bertocchi, and V. Moriggia. 1997. Postoptimality for a bond portfolio management model. In: C. Zopounidis, ed., *New Operational Approaches for Financial Modelling*, 49-62.

[60] Dupacova, J., A. Gaivoronski, Z. Kos, and T. Szantai. 1991. Stochastic programming in water management: a case study and a comparison of solution techniques. *European Journal of Operational Research* 52, 28-44.

[61] Eaves, B. C. and W. I. Zangwill. 1971. Generalized cutting plane algorithms. *SIAM Journal on Control* 9(4), 529-542.

[62] Edirisinghe, N. C. P. 1991. Essays on Bounding Stochastic Programming Problems. Ph.D. Dissertation, The University of British Columbia.

[63] Edirisinghe, N. C. P. 1996. New second-order bounds on the expectation of saddle functions with applications to stochastic linear programming. *Operations Research* 44(6), 909-922.

[64] Edirisinghe, N. C. P. 1997. Bet sizing models for Market Research, Inc., preliminary report on model formulation. University of Tennessee, Working Paper.

[65] Edirisinghe, N. C. P. 1999. Bound-based approximations in multistage stochastic programming: nonanticipativity aggregation. *Annals of Operations Research* 85, 103-127.

[66] Edirisinghe, N. C. P. 1999. Bet sizing models for Market Research, Inc., extension of the one period model, two period stochastic programming formulation. University of Tennessee, Working Paper.

[67] Edirisinghe, N. C. P., D. Atkins, and P. Iyogun. 1994. Bounds on stochastic convex allocation problems. *Applied Stochastic Models and Data Analysis* 10, 123-140.

[68] Edirisinghe, N. C. P. and G-M. You. 1996. Second-order scenario approximation and refinement in optimization under uncertainty. *Annals of Operations Research* 64, 143-178.

[69] Edirisinghe, N. C. P. and W. T. Ziemba. 1992. Tight bounds for stochastic convex programs. *Operations Research* 40(4), 660-677.

[70] Edirisinghe, N. C. P. and W. T. Ziemba. 1994. Bounds for two-stage stochastic programs with fixed recourse. *Mathematics of Operations Research* 19(2), 292-313.

[71] Edirisinghe, N. C. P. and W. T. Ziemba. 1994. Bounding the expectation of a saddle function with application to stochastic programming. *Mathematics of Operations Research* 19(2), 314-340.

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

[73] Eppen, G. D., R. K. Martin, and L. Schrage. 1989. A scenario approach to capacity planning. *Operations Research* 37(4), 517-527.

[74] EPRI. 1989. Decomposition Techniques for Multi-Area Generation and Transmission Planning Under Uncertainty. Electric Power Research Institute Report EL-6484.

[75] EPRI. 1990. Decomposition of Linear Programs Using Concurrent Processing on Multicomputers. Electric Power Research Institute Report EL-6769.

[76] Fernandez, A. A., R. L. Armacost, and J. J. Pet-Edwards. 1998. Understanding simulation solutions to resource constrained project scheduling problems with stochastic task durations. *Engineering Management Journal* 10(4), 5-13.

[77] Fine, C. H. and R. M. Freund. 1990. Optimal investment in product-flexible manufacturing capacity. *Management Science* 36(4), 449-446.

[78] Fishburn, P. C. 1977. Mean-risk analysis with risk associated with below-target returns. *The American Economic Review* 67(2), 116-126.

[79] Ford, L. R. Jr. and D. R. Fulkerson. 1958. A suggested computation for maximal multi-commodity network flows. *Management Science* 5(1), 97-101.

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

[81] Frauendorfer, K. 1996. Stochastic multistage programming in financial decision making. *Zeitschritt fur Angewandte Mathematik und Mechanik* 76, Supplement 3, 21-24.

[82] Gassmann, H. I. 1987. Multiperiod Stochastic Programming. Ph.D. Dissertation, The University of British Columbia.

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

[84] Gassmann, H. I. 1990. MSLiP user's guide. Dalhousie University, Working Paper WP-90-4.

[85] Gassmann, H. I. and A. M. Ireland. 1996. On the formulation of stochastic linear programs using algebraic modelling languages. *Annals of Operations Research* 64, 83-112.

[86] Gassmann, H. I. and E. Schweitzer. 1996. A comprehensive input format for stochastic linear programs. Dalhousie University, Working Paper WP-96-1 (revised April 1998).

[87] Geoffrion, A. M. 1970. Elements of large-scale mathematical programming part I: concepts. *Management Science* 16(11), 652-675.

[88] Geoffrion, A. M. 1970. Elements of large-scale mathematical programming part II: synthesis of algorithms and bibliography. *Management Science* 16(11), 676-691.

[89] Geoffrion, A. M. 1972. Generalized Benders decomposition. *Journal of Optimization Theory and Applications* 10(4), 237-260.

[90] Glassey, C. R. 1973. Nested decomposition and multi-stage linear programs. *Management Science* 20(3), 282-292.

[91] Glockner, G. D. and G. L. Nemhauser. 2000. A dynamic network flow problem with uncertain arc capacities: formulation and problem structure. *Operations Research* 48(2), 233-242.

[92] Glynn, P. W. and D. L. Iglehart. 1989. Importance sampling for stochastic simulations. *Management Science* 35(11), 1367-1392.

[93] Golub, G. H. and C. F. Van Loan. 1993. *Matrix Computations*, Second Edition, Fourth Printing. The Johns Hopkins University Press, Baltimore.

[94] Golub, B., M. Holmer, R. McKendall, L. Pohlman, and S. A. Zenios. 1995. A stochastic programming model for money management. *European Journal of Operational Research* 85, 282-296.

[95] Henaff, P. 1998. Hedging exotic derivatives through stochastic optimization. *Journal of Economic Dynamics and Control* 22, 1453-1466.

[96] Higle, J. L., W. W. Lowe, and R. Odio. 1994. Conditional stochastic decomposition: an algorithmic interface for optimization and simulation. *Operations Research* 42(2), 311-322.

[97] Higle, J. L. and S. Sen. 1991. Statistical verification of optimality conditions for stochastic programs with recourse. *Annals of Operations Research* 30, 215-240.

[98] Higle, J. L. and S. Sen. 1991. Stochastic decomposition: an algorithm for two-stage linear programs with recourse. *Mathematics of Operations Research* 16(3), 650-669.

[99] Hiller, R. S. and J. Eckstein. 1993. Stochastic dedication: designing fixed income portfolios using massively parallel Benders decomposition. *Management Science* 39(11), 1422-1438.

[100] Ho, J. K. and E. Loute. 1981. An advanced implementation of the Dantzig-Wolfe decomposition algorithm for linear programming. *Mathematical Programming* 20, 303-326.

[101] Ho, J. K. and E. Loute. 1983. Computational experience with advanced implementation of decomposition algorithms for linear programming. *Mathematical Programming* 27, 283-290.

[102] Ho, J. K. and A. S. Manne. 1974. Nested Decomposition for dynamic models. *Mathematical Programming* 6, 121-140.

[103] Hobbs, B. J. and Y. Ji. 1999. Stochastic programming-based bounding of expected production costs for multiarea electric power systems. *Operations Research* 47(6), 836-848.

[104] Hogan, W. W. 1973. Applications of a general convergence theory for outer approximation algorithms. *Mathematical Programming* 5, 151-168.

[105] Hsu, A. and Y. Bassok. 1999. Random yield and random demand in a production system with downward substitution. *Operations Research* 47(2), 277-290.

[106] Huang, C. C., W. T. Ziemba, and A. Ben-Tal. 1977. Bounds on the expectation of a convex function of a random variable: with applications to stochastic programming. *Operations Research* 25(2), 315-325.

[107] IBM. 1995. *Optimization Subroutine Library, Guide and Reference.*

[108] Ierapetritou, J., J. Acevedo, and E. N. Pistikopoulos. 1996. An optimization approach for process engineering problems under uncertainty. *Computers & Chemical Engineering* 20(6/7), 703-709.

[109] ILOG. 1999. *CPLEX 6.5 Reference Manual.*

[110] ILOG. 1999. *CPLEX 6.5 User's Manual.*

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

[112] Infanger, G. 1994. *Planning Under Uncertainty: Solving Large-Scale Stochastic Linear Programs*. Boyd & Fraser Publishing Company, Danvers, MA.

[113] Infanger, G. and D. P. Morton. 1996. Cut sharing for multistage stochastic linear programs with interstage dependency. *Mathematical Programming* 75, 241-256.

[114] Jacobs, J., G. Freeman, J. Grygier, D. Morton, G. Schultz, K. Staschus, and J. Stedinger. 1995. SOCRATES: a system for scheduling hydroelectric generation under uncertainty. *Annals of Operations Research* 59, 99-133.

[115] Jonsson, H., K. Jornsten, and E. A. Silver. 1993. Application of the scenario aggregation approach to a two-stage, stochastic, common component, inventory problem with a budget constraint. *European Journal of Operational Research* 68, 196-211.

[116] Jonsson, H. and E. A. Silver. 1989. Common component inventory problems with a budget constraint: heuristics and upper bounds. *Engineering Costs and Production Economics* 18, 71-81.

[117] Kall, P. 1976. *Stochastic Linear Programming*. Springer-Verlag, Berlin.

[118] Kall, P. and S. W. Wallace. 1994. *Stochastic Programming*. John Wiley & Sons, Chichester.

[119] Kato, N., M. Suzuki, S. Omachi, H. Aso, and Y. Nemoto. 1999. A handwritten character recognition system using directional element feature and asymmetric Mahalanobis distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 21(3), 258-262.

[120] Kelley, J. E. Jr. 1960. The cutting-plane method for solving convex programs. *Journal of the Society for Industrial and Applied Mathematics* 8, 703-712.

[121] Khang, D. B. and O. Fujiwara. 2000. Optimality of myopic ordering policies for inventory model with stochastic supply. *Operations Research* 48(1), 181-184.

[122] Kim, K. and J. L. Nazareth. 1991. The decomposition principle and algorithms for linear programming. *Linear Algebra and its Applications* 152, 119-133.

[123] King, A. J., R. T. Rockafellar, L. Somlyody, and R. J-B. Wets. 1988. Lake eutrophication management: the Lake Balaton project. In: Y. Ermoliev and R. J-B. Wets, eds., *Numerical Techniques for Stochastic Optimization*, 435-444.

[124] Kira, D. S. and M. I. Kusy. 1990. A stochastic capital rationing model. *Journal of the Operational Research Society* 41(7), 853-863.

[125] Kiwiel, K. C., C. H. Rosa, and A. Ruszczynski. 1999. Proximal decomposition via alternating linearization. *SIAM Journal of Optimization* 9(3), 668-669.

[126] Klaassen, P. 1998. Financial asset-pricing theory and stochastic programming models for asset/liability management: a synthesis. *Management Science* 44(1), 31-48.

[127] Konno, H. and K. Kobayashi. 1997. An integrated stock-bond portfolio optimization model. *Journal of Economic Dynamics and Control* 21, 1427-1444.

[128] Konno, H. and H. Yamazaki. 1991. Mean-absolute deviation portfolio optimization model and its applications to Tokyo stock market. *Management Science* 37(5), 519-531.

[129] Korycki, J. 1996. On a distributed implementation of a decomposition method for multistage linear stochastic programs. *Optimization* 38, 173-200.

[130] Krzanowski, W. J. 1988. *Principles of Multivariate Analysis: A User's Perspective.* Oxford University Press, New York.

[131] Kusy, M. I. and W. T. Ziemba. 1986. A bank asset and liability management model. *Operations Research* 34(3), 356-376.

[132] Lindgren, B. W. 1993. *Statistical Theory*, Fourth Edition. Chapman & Hall, New York.

[133] Louveaux, F. V. 1980. A solution method for multistage stochastic programs with recourse with application to an energy investment problem. *Operations Research* 28(4), 889-902.

[134] Louveaux, F. V. 1986. Multistage stochastic programs with block-separable recourse. In: A. Prekopa and R. J.-B. Wets, eds., *Mathematical Programming Study* 28, 48-62.

[135] Louveaux, F. V. and D. Peeters. 1992. A dual-based procedure for stochastic facility location. *Operations Research* 40(3), 564-573.

[136] Lui, Y. M. and H-D. Cheng. 1994. A new peak selection criterion based on minimizing the classification error. *Information Sciences* 94, 213-233.

[137] Lustig, I. J., J. M. Mulvey, and T. J. Carpenter. 1991. Formulating two-stage stochastic programs for interior point methods. *Operations Research* 39(5), 757-770.

[138] Madansky, A. 1963. Dual variables in two-stage linear programming under uncertainty. *Journal of Mathematical Analysis and Applications* 6, 98-108.

[139] Magnus, J. R. and H. Neudecker. 1988. *Matrix Differential Calculus with Applications in Statistics and Econometrics*. John Wiley & Sons, Chichester.

[140] Markowitz, H. 1952. Portfolio selection. *The Journal of Finance* 7, 77-91.

[141] Markowitz, H. 1952. The utility of wealth. *The Journal of Political Economy* 60, 151-158.

[142] Markowitz, H. M. 1959. *Portfolio Selection: Efficient Diversification of Investments*. Yale University Press, New Haven.

[143] Martel, A., R. Nadeau, and W. L. Price. 1990. A stochastic programming approach to the estimation of shell fragmentation distributions. *INFOR* 28(1), 53-66.

[144] Messina, E. and G. Mitra. 1997. Modelling and analysis of multistage stochastic programming problems: a software environment. *European Journal of Operational Research* 101, 343-359.

[145] Meszaros, C. 1997. The augmented system variant of IPMs in two-stage stochastic linear programming computation. *European Journal of Operational Research* 101, 317-327.

[146] Microsoft. 1996. *Microsoft Windows NT Workstation Resource Kit*.

[147] Morton, D. P. 1996. An enhanced decomposition algorithm for multistage stochastic hydroelectric scheduling. *Annals of Operations Research* 64, 211-235.

[148] Morton, D. P. 1998. Stopping rules for a class of sampling-based stochastic programming algorithms. *Operations Research* 46(5), 710-718.

[149] Morton, D. P. and R. K. Wood. 1999. Restricted-recourse bounds for stochastic linear programming. *Operations Research* 47(6), 943-956.

[150] Mossin, J. 1968. Optimal multiperiod portfolio policies. *The Journal of Business* 41, 215-229.

[151] Mulvey, J. M., D. P. Rosenbaum, and B. Shetty. 1997. Strategic financial risk management and operations research. *European Journal of Operational Research* 97, 1-16.

[152] Mulvey, J. M. and A. Ruszczynski. 1992. A new scenario decomposition method for large-scale stochastic optimization. Princeton University, Technical Report SOR-91-19.

[153] Mulvey, J. M. and A. Ruszczynski. 1995. A new scenario decomposition method for large-scale stochastic optimization. *Operations Research* 43(3), 477-490.

[154] Mulvey, J. M. and H. Vladimirou. 1992. Stochastic network programming for financial planning problems. *Management Science* 38(11), 1642-1664.

[155] Murty, K. G. 1968. Linear programming under uncertainty: a basic property of the optimal solution. *Zeitschrift fur Wahrscheinlichkeitstheorie und verwandte Gebiete* 10, 284-288.

[156] National Bureau of Standards. 1972. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, Tenth Printing. M. Abramowitz and I. A. Stegun, editors. United States Government Printing Office, Washington, D. C.

[157] Nazareth, J. L. 1987. *Computer Solutions of Linear Programs*. Oxford University Press, New York.

[158] Nielsen, S. S. and S. A. Zenios. 1997. Scalable parallel Benders decomposition for stochastic linear programming. *Parallel Computing* 23, 1069-1088.

[159] Noble, B. and J. W. Daniel. 1988. *Applied Linear Algebra*. Prentice-Hall, Englewood Cliffs.

[160] Olsen, P. 1976. When is a multistage stochastic programming problem well-defined? *SIAM Journal on Control and Optimization* 14(3), 518-527.

[161] Olsen, P. 1976. Multistage stochastic programming with recourse as mathematical programming in an L_p Space. *SIAM Journal on Control and Optimization* 14(3), 528-537.

[162] Paules, G. E. IV and C. A. Floudas. 1992. Stochastic programming in process synthesis: a two-stage model with MINLP recourse for multiperiod heat-integrated distillation sequences. *Computers & Chemical Engineering* 16(3), 189-210.

[163] Penny, K. I. 1996. Appropriate critical values when testing for a single multivariate outlier by using the Mahalanobis distance. *Applied Statistics* 45(1), 73-81.

[164] Pereira, M. V. F. and L. M. V. G. Pinto. 1985. Stochastic optimization of a multireservoir hydroelectric system: a decomposition approach. *Water Resources Research* 21(6), 779-792.

[165] Pereira, M. V. F. and L. M. V. G. Pinto. 1991. Multi-stage stochastic optimization applied to energy planning. *Mathematical Programming* 52, 359-375.

[166] Pieptea, D. R. 1987. Leveraged bond portfolio optimization under uncertainty. *The Financial Review* 22(1), 87-109.

[167] Pistikopoulos, E. N. and M. G. Ierapetritou. 1995. Novel approach for optimal process design under uncertainty. *Computers & Chemical Engineering* 19(10), 1089-1110.

[168] Pistikopoulos, E. N., T. V. Thomaidis, A. Melin, and M. G. Ierapetritou. 1996. Flexibility, reliability and maintenance considerations in batch plant design under uncertainty. *Computers & Chemical Engineering* 20 Supplement, S1209-S1214.

[169] Prekopa, A. 1995. *Stochastic Programming*. Kluwer Academic Publishers, Dordrecht, The Netherlands.

[170] Prisman, E. Z., M. B. Slovin, and M. E. Sushka. 1986. A general model of the banking firm under conditions of monopoly, uncertainty, and recourse. *Journal of Monetary Economics* 17, 293-304.

[171] Qi, L. 1985. Forest iteration method for stochastic transportation problem. In: R. W. Cottle, ed., *Mathematical Programming Study* 25, 142-163.

[172] Qiu, J. and A. A. Girgis. 1993. Optimization of power system reliability level by stochastic programming. *Electric Power Systems Research* 26(2), 87-95.

[173] Rencher, A. C. 1995. *Mehtods of Multivariate Analysis*. John Wiley & Sons, Inc., New York.

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

[175] Romisch, W. and R. Schultz. 1996. Decomposition of a multi-stage stochastic program for power dispatch. *Zeitschrift fur Angewandte Mathematik und Mechanik* 76 Supplement 3, 29-32.

[176] Rosa, C. H. and A. Ruszczynski. 1996. On augmented Lagrangian decomposition methods for multistage stochastic programs. *Annals of Operations Research* 64, 289-309.

[177] Rosa, C. H. and S. Takriti. 1999. Improving aggregation bounds for two-stage stochastic programs. *Operations Research Letters* 24, 127-137.

[178] Rotting, T. A. and A. Gjelsvik. 1992. Stochastic dual dynamic programming for seasonal scheduling in the Norweigian Power System. *IEEE Transactions on Power Systems* 7(1), 273-279.

[179] Ruszczynski, A. 1986. A regularized decomposition method for minimizing a sum of polyhedral functions. *Mathematical Programming* 35, 309-333.

[180] Ruszczynski, A. 1993. Parallel decomposition of multistage stochastic programming problems. *Mathematical Programming* 58, 201-228.

[181] Ruszczynski, A. 1995. On convergence of an augmented Lagrangian decomposition method for sparse convex optimization. *Mathematics of Operations Research* 20(3), 634-656.

[182] Ruszczynski, A. 1995. On the regularized decomposition method for stochastic programming problems. In: K. Marti and P. Kall, eds., *Proceedings of the 2nd GAMM/IFIP – Workshop on "Stochastic Optimization: Numerical Methods and Technical Applications"*, 93-108.

[183] Ruszczynski, A. 1997. Decomposition methods in stochastic programming. *Mathematical Programming* 79, 333-353.

[184] Ruszczynski, A. and A. Swietanowski. 1997. Accelerating the regularized decomposition method for two stage stochastic linear programs. *European Journal of Operational Research* 101, 328-342.

[185] Sanghvi, A. P. and I. H. Shavel. 1986. Investment planning for hydro-thermal power system expansion: stochastic programming employing the Dantzig-Wolfe decomposition principle. *IEEE Transactions on Power Systems* PWRS-1(2), 115-121.

[186] Sapountzis, C. 1989. Allocating blood to hospitals. *Journal of the Operational Research Society* 40(5), 443-449.

[187] Sharpe, W. F. 1966. Mutual Fund Performance. *The Journal of Business* 39, 119-138.

[188] Sharpe, W. F. 1985. *Investments*, Third Edition.Prentice Hall, Inc., Englewood Cliffs.

[189] Silva, E. L., M. Morozowski, L. G. S. Fonseca, G. C. Oliveira, A. C. G. Melo, and J. C. O. Mello. 1995. Transmission constrained maintenance scheduling of generating units: a stochastic programming approach. *IEEE Transactions on Power Systems 10(2), 695-701.*

[190] Sinha, D. and J. C. Wei. 1992. Stochastic analysis of flexible process choices. *European Journal of Operational Research* 60, 183-199.

[191] Somlyody, L. and R. J.-B. Wets. 1988. Stochastic optimization models for lake eutrophication management. *Operations Research* 36(5), 660-681.

[192] Takriti, S., B. Krasenbrink, and L. S.-Y. Wu. 2000. Incorporating fuel constraints and electricity spot prices into the stochastic unit commitment problem. *Operations Research* 48(2), 268-280.

[193] Taylor, A. E. and W. R. Mann. 1983. *Advanced Calculus*, Third Edition. John Wiley & Sons, Inc., New York.

[194] Taylor, H. M. and S. Karlin. 1994. *An Introduction to Stochastic Modeling*, Revised Edition. Academic Press, San Diego.

[195] Van Slyke, R. and R. Wets. 1966. Programming under uncertainty and stochastic optimal control. *SIAM Journal on Control* 4(1), 179-193.

[196] Van Slyke, R. M. and R. Wets. 1969. L-Shaped linear programs with applications to optimal control and stochastic programming. *SIAM Journal on Applied Mathematics* 17(4), 638-663.

[197] Vladimirou, H. 1998. Computational assessment of distributed decomposition methods for stochastic linear programs. *European Journal of Operational Research* 108, 653-670.

[198] Vladimirou, H. and S. A. Zenios. 1999. Scalable parallel computations for large-scale stochastic programming. *Annals of Operations Research* 90, 87-129.

[199] Van der Vlerk, Maarten H. 2001. *Stochastic Programming Bibliography*. World Wide Web, http://mally.eco.rug.nl/biblio/stoprog.html.

[200] Wagner, J. M., U. Shamir, and D. H. Marks. 1994. Containing groundwater contamination: planning models using stochastic programming with recourse. *European Journal of Operational Research* 77, 1-26.

[201] Watanabe, T. and H. Ellis. 1993. Stochastic programming models for air quality management. *Computers & Operations Research* 20(6), 651-663.

[202] Wets, R. J. B. 1966. Programming under uncertainty: the equivalent convex program. *SIAM Journal on Applied Mathematics* 14(1), 89-105.

[203] Wets, R. 1966. Programming under uncertainty: the solution set. *SIAM Journal on Applied Mathematics* 14(5), 1143-1151.

[204] Wets, R. 1966. Programming under uncertainty: the complete problem. *Zeitschrift fur Wahrscheinlichkeitstheorie und verwandte Gebiete* 4, 316-339.

[205] Wets, R. J-B. 1974. Stochastic programming with fixed recourse: the equivalent deterministic program. *SIAM Review* 16(3), 309-339.

[206] Wets, R. 1983. Stochastic programming: solution techniques and approximation schemes. In: A. Bachem, M. Grotschel, and B. Korte, eds., *Mathematical Programming: The State of the Art, Bonn* 1982, 566-603.

[207] Wittrock, R. J. 1985. Dual nested decomposition of staircase linear programs. In: R. W. Cottle, ed., *Mathematical Programming Study* 24, 65-86.

[208] Wollmer, R. D. 1985. Critical path planning under uncertainty. In: R. W. Cottle, ed., *Mathematical Programming Study* 25, 164-171.

[209] Wood, M. K. and G. B. Dantzig. 1949. Programming of independent activities I: general discussion. *Econometrica* 17(3-4), 193-199.

[210] Worzel, K. J., C. Vassiadou-Zeniou, and S. A. Zenios. 1994. Integrated simulation and optimization models for tracking indices of fixed-income securities. *Operations Research* 42(2), 223-233.

[211] Yakowitz, D. 1992. An exact penalty algorithm for recourse-constrained stochastic linear programs. *Applied Mathematics and Computation* 49, 39-62.

[212] Yakowitz, D. S. 1994. A regularized stochastic decomposition algorithm for two-stage stochastic linear programs. *Computational Optimization and Applications* 3, 59-81.

[213] Zakeri, G., A. B. Philpott, and D. M. Ryan. 2000. Inexact cuts in Benders decomposition. *SIAM Journal on Optimization* 10(3), 643-657.

[214] Zenios, S. A., M. R. Holmer, R. McKendall, and C. Vassiadou-Zeniou. 1998. Dynamic models for fixed-income portfolio management under uncertainty. *Journal of Economic Dynamics and Control* 22, 1517-1541.

[215] Ziemba, W. T. 1970. Computational algorithms for convex stochastic programs with simple recourse. *Operations Research* 18(3), 414-431.

APPENDICES

Appendix A
Equivalence of Node Labeling Schemes

Derivations and proofs for the equivalence of the path vector and period-index node labeling schemes developed in Chapter 2 are given below. Equivalence of the two schemes is extremely important since they are frequently used together in problem formulations. Both schemes assume the use of the convention that all nodes in a given period have the same number of possible outcomes as described in Section 2.2.1. The derivation and validity of equation [2.6] on page 28 for determining the period-index label given the path vector are discussed in the first section below. The second section contains the derivation for equation [2.7] on page 28 for determining the path vector given the period-index label. Relating the two labeling schemes when the decision tree is traversed in a systematic manner is the topic of the third and final section of this appendix.

A.1 Path Vector to Period-Index

Equation [2.6],

$$h_t = \begin{cases} 1, & \text{if } t = 1, \\ l_{t-1} + \sum_{j=1}^{t-2} (l_j - 1)\, L_{j+1} L_{j+2} \cdots L_{t-1}, & \text{if } 2 \le t \le T, \end{cases}$$

is used to find the period-index label (t, h_t) for a node in period t, $1 \le t \le T$, given the path vector $[\bullet]_t = [l_1, \ldots, l_{t-1}]$ to that node. The single first period node represented by the null or empty path vector, $[\,]$, clearly has a period-index label of $(1, 1)$. A simple example helps to visualize the derivation of the equation for a node in a period later than the first.

Appendix A Equivalence of Node Labeling Schemes

Figure A.1 shows the path vector $[2, 2, 2]$ to the last node in a four-period binary outcomes decision tree as the darkened path through the tree. The within period sequential index, $h_4 = 8$, for this node is a function of the number of nodes above it in the tree. There is a group of nodes on or above node $[2, 2, 2]$ for each of its three ancestor nodes along the path. Group one consists only of four descendents of the first period node. The second group contains two descendents of the grandparent node in the second period while the final group is the older child of the parent node. The number of nodes in each of the first two groups is the index for the outcome associated with the corresponding ancestor node less one times the product of the number of outcomes possible in the periods between the ancestor node and node $[2, 2, 2]$. Summing these products and the corresponding outcome index for the parent node will give the sequential index for the node in question:

$$\begin{aligned} h_4 &= (l_1 - 1)\, L_2 L_3 + (l_2 - 1)\, L_3 + l_3 \\ &= (2-1)(2)(2) + (2-1)(2) + 2 \\ &= 4 + 2 + 2 \\ &= 8. \end{aligned}$$

The above discussion suggests that the within period sequential node index, h_t, for a node with a path vector of $[l_1, \ldots, l_{t-1}]$ in period t, $2 \leq t \leq T$, is

$$\begin{aligned} h_t &= (l_1 - 1)\, L_2 L_3 \cdots L_{t-1} + (l_2 - 1)\, L_3 \cdots L_{t-1} + \ldots + (l_{t-2} - 1)\, L_{t-1} + l_{t-1} \\ &= l_{t-1} + \sum_{j=1}^{t-2} (l_j - 1)\, L_{j+1} L_{j+2} \cdots L_{t-1} \end{aligned}$$

which agrees with equation [2.6]. Induction is used below to show that equation [2.6] is valid in the general case.

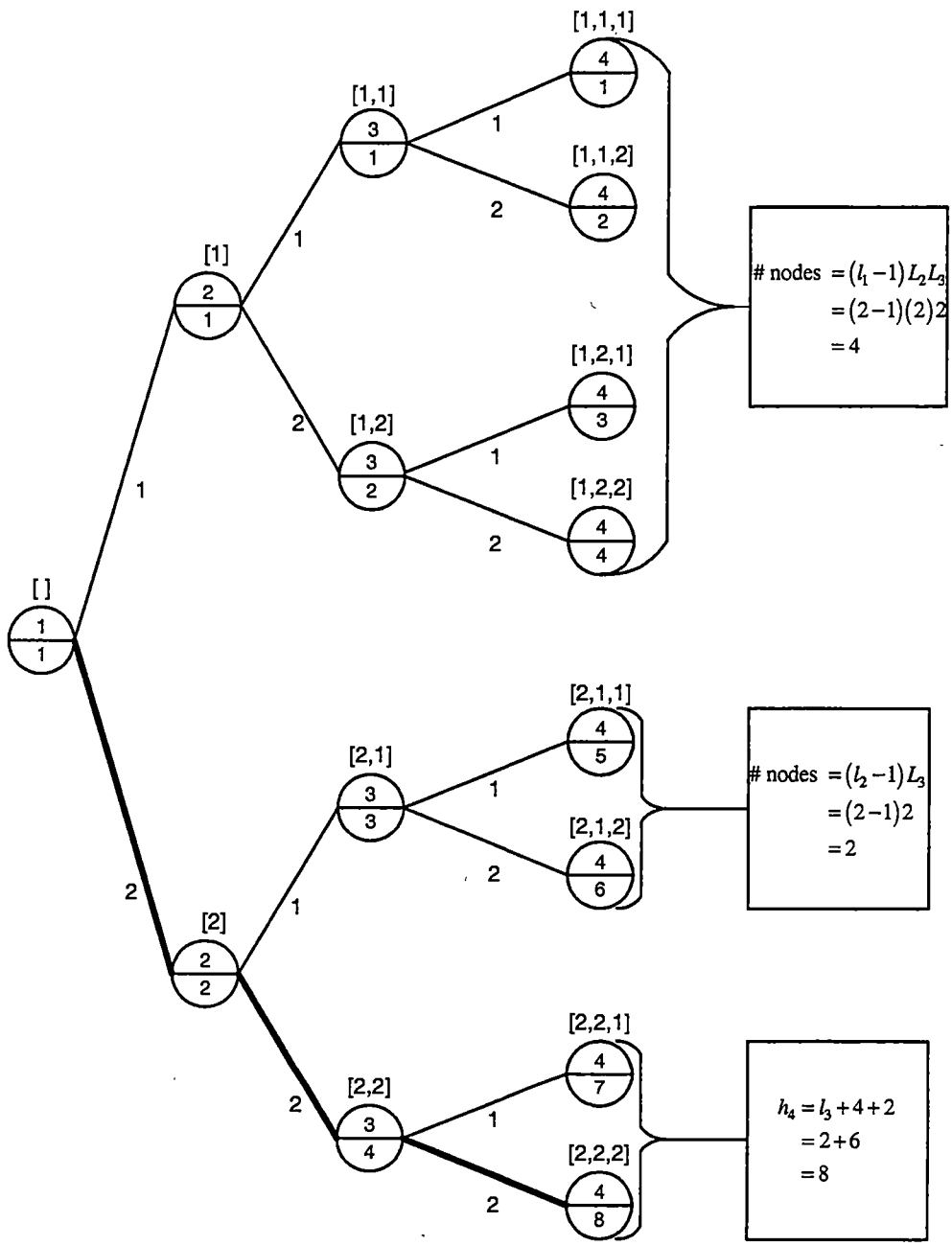


Figure A.1: Equivalence of Labels for Node $[2, 2, 2] = (4, 8)$ In a Four-Period Binary Outcomes Decision Tree

Equation [2.6] is clearly valid for the first period and also for the second period since $h_2 = l_1$ and the summation term in equation [2.6] is zero. Assume the equation is valid for any node in period $t-1$ for $3 \leq t \leq T$. Then given the path vector, $[l_1, l_2, \ldots, l_{t-2}, l_{t-1}]$, to a node in period t, the h_{t-1} in the period-index label $(t-1, h_{t-1})$ for the parent node $[l_1, l_2, \ldots, l_{t-2}]$ in period $t-1$ is

$$h_{t-1} = l_{t-2} + \sum_{j=1}^{t-3} (l_j - 1)\, L_{j+1} L_{j+2} \cdots L_{t-2}.$$

Each node $(t-1, k)$, $k = 1, \ldots, h_{t-1} - 1$, in period $t-1$ has L_{t-1} child nodes in period t that are above node $[l_1, l_2, \ldots, l_{t-2}, l_{t-1}]$ which is child number l_{t-1} of its parent node implying that

$$\begin{aligned} h_t &= \left(-1 + l_{t-2} + \sum_{j=1}^{t-3} (l_j - 1)\, L_{j+1} L_{j+2} \cdots L_{t-2} \right) L_{t-1} + l_{t-1} \\ &= l_{t-1} + (l_{t-2} - 1)\, L_{t-1} + \sum_{j=1}^{t-3} (l_j - 1)\, L_{j+1} L_{j+2} \cdots L_{t-2} L_{t-1} \\ &= l_{t-1} + \sum_{j=1}^{t-2} (l_j - 1)\, L_{j+1} L_{j+2} \cdots L_{t-2} L_{t-1}. \end{aligned}$$

Therefore, equation [2.6] is valid by inductive reasoning.

A.2 Period-Index to Path Vector

Equation [2.7],

$$l_k = 1 + \left\lfloor \frac{h_t - 1 - \sum_{j=1}^{k-1} (l_j - 1) L_{j+1} L_{j+2} \cdots L_{t-1}}{\prod_{j=k+1}^{t-1} L_j} \right\rfloor = \left\lceil \frac{h_t - \sum_{j=1}^{k-1} (l_j - 1) L_{j+1} L_{j+2} \cdots L_{t-1}}{\prod_{j=k+1}^{t-1} L_j} \right\rceil$$

is solved in sequential order for $k = 1, \ldots, t-1$ to determine the path vector $[l_1, \ldots, l_{t-1}]$ to a node in period t given the period-index label (t, h_t) for that node. Either right-hand-side term of equation [2.7] may be used. Equation [2.6] is used to derive the above relationship.

Assume that the values $l_1, \ldots, l_{k-1}$ are known for some k such that $2 \leq k \leq t-1$. Then equation [2.6] implies that

$$\begin{aligned} h_t &= l_{t-1} + \sum_{j=1}^{k-1} (l_j - 1)\, L_{j+1} \cdots L_{t-1} + (l_k - 1)\, L_{k+1} \cdots L_{t-1} \\ &\quad + \sum_{j=k+1}^{t-2} (l_j - 1)\, L_{j+1} \cdots L_{t-1}. \end{aligned}$$

Rearranging the above equation yields

$$(l_k - 1) \prod_{j=k+1}^{t-1} L_j + l_{t-1} + \sum_{j=k+1}^{t-2} (l_j - 1)\, L_{j+1} \cdots L_{t-1} = h_t - \sum_{j=1}^{k-1} (l_j - 1)\, L_{j+1} \cdots L_{t-1}, \qquad [A.1]$$

and

$$l_{t-1} + \sum_{j=k+1}^{t-2} (l_j - 1)\, L_{j+1} \cdots L_{t-1} > 0 \qquad [A.2]$$

since $L_t \geq 1$ and $1 \leq l_t \leq L_t$ for $t = 1, ..., T$. Removing the term on the left of inequality [A.2] from equation [A.1] implies that

$$l_k - 1 < \frac{h_t - \sum_{j=1}^{k-1} (l_j - 1) L_{j+1} L_{j+2} \cdots L_{t-1}}{\prod_{j=k+1}^{t-1} L_j}$$

$$\Longrightarrow \quad l_k - 1 < \left\lceil \frac{h_t - \sum_{j=1}^{k-1} (l_j - 1) L_{j+1} L_{j+2} \cdots L_{t-1}}{\prod_{j=k+1}^{t-1} L_j} \right\rceil .$$

Then, since l_k is an integer,

$$l_k \leq \left\lceil \frac{h_t - \sum_{j=1}^{k-1} (l_j - 1) L_{j+1} L_{j+2} \cdots L_{t-1}}{\prod_{j=k+1}^{t-1} L_j} \right\rceil . \qquad \text{[A.3]}$$

Subtracting one from both sides of equation [A.1] results in

$$\begin{aligned} & (l_k - 1) \prod_{j=k+1}^{t-1} L_j + l_{t-1} - 1 + \sum_{j=k+1}^{t-2} (l_j - 1) L_{j+1} \cdots L_{t-1} \\ = \; & h_t - 1 - \sum_{j=1}^{k-1} (l_j - 1) L_{j+1} \cdots L_{t-1}, \end{aligned}$$

and dividing through the above equation by $\prod_{j=k+1}^{t-1} L_j$ gives

$$l_k - 1 + \frac{l_{t-1} - 1 + \sum_{j=k+1}^{t-2} (l_j - 1) L_{j+1} \cdots L_{t-1}}{\prod_{j=k+1}^{t-1} L_j} = \frac{h_t - 1 - \sum_{j=1}^{k-1} (l_j - 1) L_{j+1} \cdots L_{t-1}}{\prod_{j=k+1}^{t-1} L_j},$$

which implies that

$$l_k - 1 + \frac{l_{t-1} - 1 + \sum_{j=k+1}^{t-2} (l_j - 1) L_{j+1} \cdots L_{t-1}}{\prod_{j=k+1}^{t-1} L_j} \geq \left\lfloor \frac{h_t - 1 - \sum_{j=1}^{k-1} (l_j - 1) L_{j+1} \cdots L_{t-1}}{\prod_{j=k+1}^{t-1} L_j} \right\rfloor . \qquad \text{[A.4]}$$

Now, since the left-hand-side of inequality [A.2] is an integer, subtracting one from the left-hand-side of that inequality implies that

$$l_{t-1} - 1 + \sum_{j=k+1}^{t-2} (l_j - 1) L_{j+1} \cdots L_{t-1} \geq 0. \qquad \text{[A.5]}$$

The relations $1 \leq l_t \leq L_t$ for $t = 1, \ldots, T$ imply that

$$l_{t-1} - 1 + \sum_{j=k+1}^{t-2} (l_j - 1) L_{j+1} \cdots L_{t-1} \leq L_{t-1} - 1 + \sum_{j=k+1}^{t-2} (L_j - 1) L_{j+1} \cdots L_{t-1}. \qquad \text{[A.6]}$$

Let $\hat{L}$ equal the right-hand-side of inequality [A.6], then

$$\begin{aligned}
\hat{L} &= L_{t-1} - 1 + \sum_{j=k+1}^{t-2} (L_j - 1) L_{j+1} \cdots L_{t-1} \\
&= L_{t-1} - 1 + \sum_{j=k+1}^{t-2} \left(\prod_{i=j}^{t-1} L_i \right) - \sum_{j=k+1}^{t-2} \left(\prod_{i=j+1}^{t-1} L_i \right) \\
&= L_{t-1} - 1 + \prod_{j=k+1}^{t-1} L_j + \sum_{j=k+2}^{t-2} \left(\prod_{i=j}^{t-1} L_i \right) - \sum_{j=k+1}^{t-3} \left(\prod_{i=j+1}^{t-1} L_i \right) - L_{t-1} \\
&= -1 + \prod_{j=k+1}^{t-1} L_j + \sum_{j=k+2}^{t-2} \left(\prod_{i=j}^{t-1} L_i \right) - \sum_{j=k+2}^{t-2} \left(\prod_{i=j}^{t-1} L_i \right) \\
\hat{L} &= -1 + \prod_{j=k+1}^{t-1} L_j.
\end{aligned}$$

Then inequality [A.6] becomes

$$l_{t-1} - 1 + \sum_{j=k+1}^{t-2} (l_j - 1) L_{j+1} \cdots L_{t-1} \leq -1 + \prod_{j=k+1}^{t-1} L_j,$$

or, equivalently,

$$l_{t-1} - 1 + \sum_{j=k+1}^{t-2} (l_j - 1) L_{j+1} \cdots L_{t-1} < \prod_{j=k+1}^{t-1} L_j. \qquad \text{[A.7]}$$

Since $L_t \geq 1$ for $t = 1, \ldots, T$, inequalities [A.5] and [A.7] imply that

$$0 \leq \frac{l_{t-1} - 1 + \sum_{j=k+1}^{t-2} (l_j - 1) L_{j+1} \cdots L_{t-1}}{\prod_{j=k+1}^{t-1} L_j} < 1,$$

and since the center term above is the fractional term in the left-hand-side of inequality [A.4], the latter inequality may be simplified to

$$l_k \geq 1 + \left\lfloor \frac{h_t - 1 - \sum_{j=1}^{k-1} (l_j - 1) L_{j+1} \cdots L_{t-1}}{\prod_{j=k+1}^{t-1} L_j} \right\rfloor . \qquad \text{[A.8]}$$

Inequalities [A.3] and [A.8] establish respectively upper and lower bounds on l_k. It can be shown, however, that the right-hand-sides of inequalities [A.3] and [A.8] are equal. Define integer m equal to the right-hand-side of inequality [A.3]:

$$m = \left\lceil \frac{h_t - \sum_{j=1}^{k-1} (l_j - 1) L_{j+1} L_{j+2} \cdots L_{t-1}}{\prod_{j=k+1}^{t-1} L_j} \right\rceil ,$$

then

$$m - 1 < \frac{h_t - \sum_{j=1}^{k-1} (l_j - 1) L_{j+1} L_{j+2} \cdots L_{t-1}}{\prod_{j=k+1}^{t-1} L_j} \leq m,$$

or, equivalently,

$$(m-1) \prod_{j=k+1}^{t-1} L_j < h_t - \sum_{j=1}^{k-1} (l_j - 1) L_{j+1} L_{j+2} \cdots L_{t-1} \leq m \prod_{j=k+1}^{t-1} L_j$$

since $\prod_{j=k+1}^{t-1} L_j \geq 1$. Since all three terms in the above expression are integers, subtracting one from the middle term yields

$$(m-1)\prod_{j=k+1}^{t-1} L_j \leq h_t - 1 - \sum_{j=1}^{k-1}(l_j - 1)L_{j+1}L_{j+2}\cdots L_{t-1} < m\prod_{j=k+1}^{t-1} L_j,$$

which implies that

$$m-1 = \left\lfloor \frac{h_t - 1 - \sum_{j=1}^{k-1}(l_j - 1)L_{j+1}L_{j+2}\cdots L_{t-1}}{\prod_{j=k+1}^{t-1} L_j} \right\rfloor,$$

or

$$1 + \left\lfloor \frac{h_t - 1 - \sum_{j=1}^{k-1}(l_j - 1)L_{j+1}L_{j+2}\cdots L_{t-1}}{\prod_{j=k+1}^{t-1} L_j} \right\rfloor = \left\lceil \frac{h_t - \sum_{j=1}^{k-1}(l_j - 1)L_{j+1}L_{j+2}\cdots L_{t-1}}{\prod_{j=k+1}^{t-1} L_j} \right\rceil. \quad \text{[A.9]}$$

Therefore, inequalities [A.3] and [A.8] and equation [A.9] result in

$$l_k = 1 + \left\lfloor \frac{h_t - 1 - \sum_{j=1}^{k-1}(l_j - 1)L_{j+1}L_{j+2}\cdots L_{t-1}}{\prod_{j=k+1}^{t-1} L_j} \right\rfloor = \left\lceil \frac{h_t - \sum_{j=1}^{k-1}(l_j - 1)L_{j+1}L_{j+2}\cdots L_{t-1}}{\prod_{j=k+1}^{t-1} L_j} \right\rceil$$

so that equation [2.7] will determine the path vector given the period-index label.

A.3 Practical Implementation

Equations [2.6] and [2.7] would be useful in relating path vector and period-index labels when nodes are selected on a random basis. In most practical situations, however, the decision tree is traversed in a systematic manner – generally in an iterative, breadth-first,

or a recursive, depth-first, node order. Use of equations [2.6] and [2.7] would be both inefficient and unnecessary in these situations.

Path vectors are used to traverse all decision trees discussed in this thesis. The path vector scheme is the ideal way to label the nodes when a recursive tree traversal is used. A practical implementation of recursive tree traversal is described in the first subsection below. Period-index labels, on the other hand, are more practical when the tree is traversed in an iterative order. Implementing a practical period-index to path vector translation in an iterative tree traversal without the use of equation [2.7] is the focus of the second subsection.

The following notation, in addition to that established in Section 2.2.1, will be used in the following discussions. A path vector is denoted by $path$ and element j will be designated as $path\,(j)$. The colon notation of such programming languages as Fortran 95 and MATLAB[7] is used to designate a particular array section. Array colon notation can be summarized as: $path(j:k)$ is the vector formed by elements $j, j+1, \ldots, k-1, k$ where $path(j:k) = [\,]$ if $k < j$ or the indices are otherwise invalid. Note that $path(1:0) = [\,]$ is a null or empty vector and designates the first period root node of the tree. Also note that $path(j:j)$ is a vector with a single element (or a null vector if j is an invalid index) and not a scalar. The path vector is assumed to be allocated size of at least $T-1$ so that the elements with indices $1, \ldots, T-1$ are accessible.

[7] MATLAB is a registered trademark of The MathWorks, Inc., Natick, MA

A.3.1 Recursive Tree Traversal

Recursive tree traversal is performed in a depth-first node order as indicated by the bold and underlined node indices to the left of each node in Figure A.2. The recursive function pseudo-code below is one way to efficiently implement a depth-first node order traversal. Argument t is the index for the current period and must be set to one on the initial entry (i.e., on the call from the driver routine), T is the number of periods represented in the tree, and *path* is the allocated path vector. The ellipsis in the argument list represents any remaining arguments that would be required by the function. The number of outcomes, L_t, possible at any node in period t, $1 \leq t \leq T-1$, is assumed to be available through shared memory (such as global in C/C++ or a module in Fortran 95).

```
recursive function( t, T, path, ...)
   access the node pointed to by path(1 : t − 1)
          ⋮
   executable command statements
          ⋮
   if (t = T) return
   for l_t = 1, ..., L_t
      path(t) = l_t
      call recursive function ((t + 1), T, path, ...)
   end for
end recursive function
```

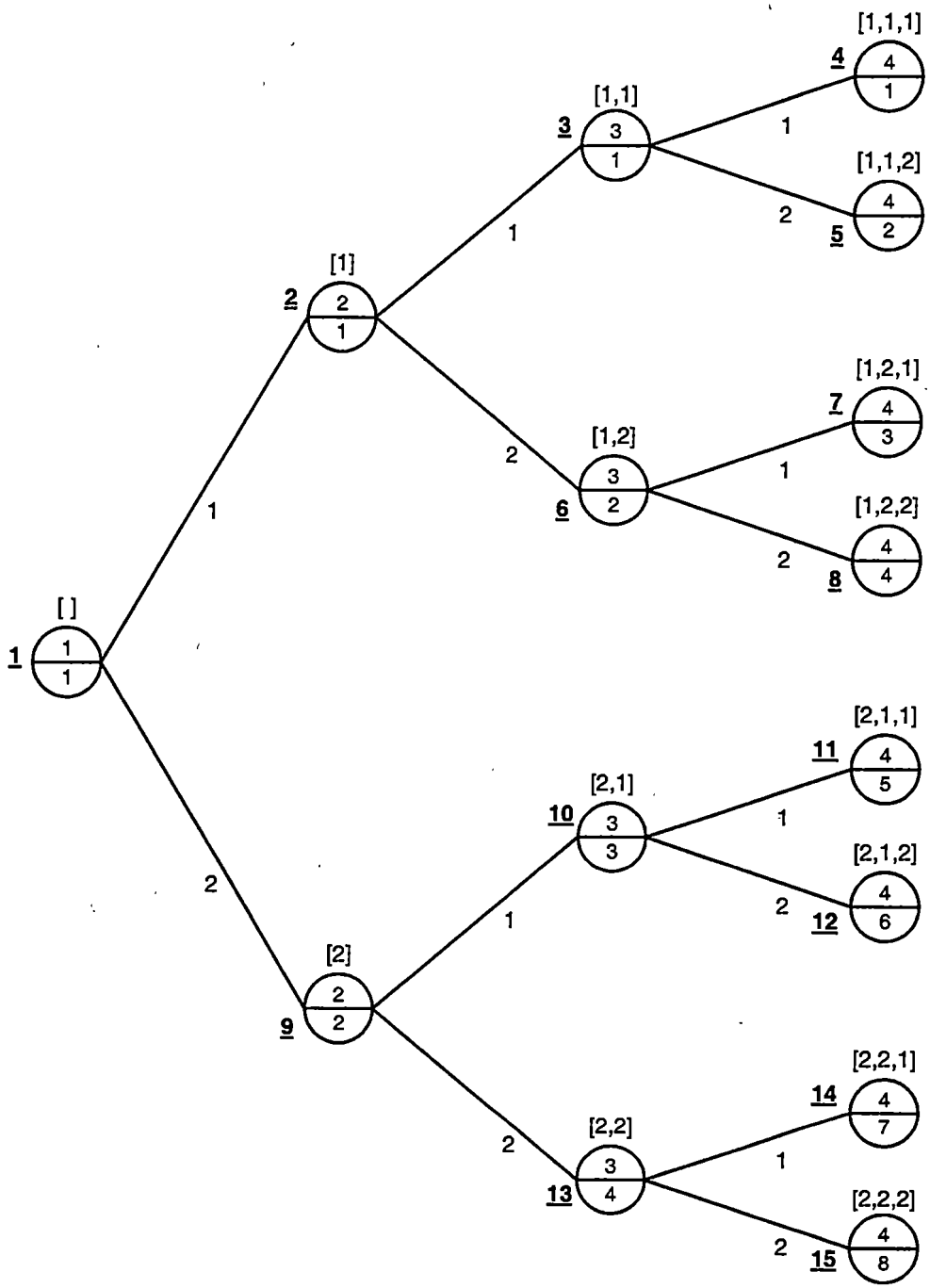


Figure A.2: Decision Tree Node Indices in a Recursive, Depth-First, Order

The *if* ($t = T$) *return* statement is the terminal condition of the recursion.

A.3.2 Iterative Tree Traversal

Iterative tree traversal is performed in a breadth-first node order as indicated by the bold and underlined node indices to the left of each node in Figure A.3. The pseudo-code block below is one way to efficiently implement a breadth-first node order traversal. The number of periods, T, and the number of outcomes, L_t, possible at each node in periods $t = 1, \ldots, T-1$ are assumed to be accessible.

```
    ⋮
access the first period node pointed to by path(1:0)
    ⋮
executable command statements
    ⋮
H = 1
for t = 2,...,T
    j = t - 1
    k = j
    path(1:j) = 1
    path(j) = 0
    H = H * L_{t-1}
    for h = 1,...,H
```

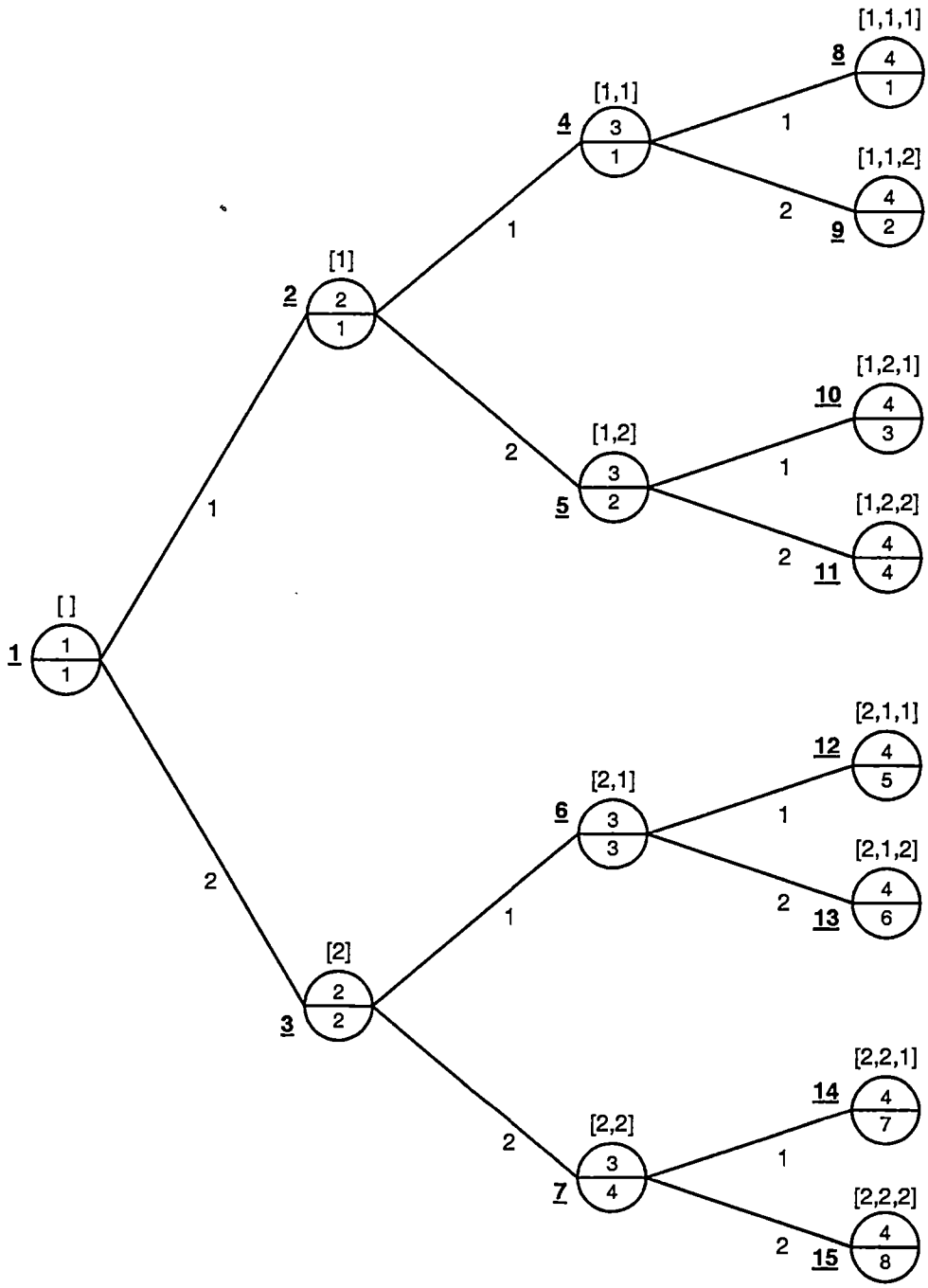


Figure A.3: Decision Tree Node Indices in an Iterative, Breadth-First, Order

```
do
    path(k) = path(k) + 1
    if ( path(k) ≤ L_t ) exit do
    path(k) = 1
    k = k − 1
end do
k = j
access the node pointed to by path(1 : t − 1)
    ⋮
executable command statements
    ⋮
  end for
end for
```

Note that the desired process statements for the data associated with the first period node are executed outside the *for* loops to preclude an attempt to access an invalid index of *path*, i.e., *path* (0) in the outer loop.

Appendix B
Algorithm MDPCA Listings

Detailed listings are provided below for the myopic dual-primal cycling algorithm, **MDPCA**, and four procedures referenced by the algorithm. These listings are based upon properties of the algorithm that are established in Section 4.5.2 starting on page 155. Also recall that the algorithm assumes a problem that is bounded and has complete or relatively complete recourse. Algorithm **MDPCA** is listed first followed in order by the referenced procedures **Initialize(Duals Lead)**, **Initialize(Primals Lead)**, **Solve(Duals Lead)**, and **Solve(Primals Lead)**. Flowcharts for each procedure follow the listings. This appendix concludes with some remarks concerning possible modifications to the algorithm.

Note that the path vector, $[\bullet]_t = [l_1, \ldots, l_{t-1}]$, and period-index, (t, h_t), node labeling schemes described in Section 2.2.1 starting on page 24 are used interchangeably in the listings (with the period identification subscript on h_t omitted). Equation [2.7] on page 28 can be used to determine the path vector given the period-index label for a particular node. The path vector is required in order to obtain stored stochastic data as described in Section 4.6. In practice, the procedures outlined in section A.3 are used to relate the two labeling schemes and obtain the necessary data.

B.1 Algorithm MDPCA

This listing assumes that the desired version, duals lead or primals lead, and the maximum desired relative bounds gap, ϵ, are prespecified. Explanatory remarks follow the listing.

Algorithm MDPCA

Step 0: Initialize the following parameters:

$$\begin{aligned} k &\leftarrow 0, \\ z_{\text{LB}} &\leftarrow -\infty, \\ z_{\text{UB}} &\leftarrow \infty, \\ diff &\leftarrow \infty. \end{aligned}$$

Go to Step 1.

Step 1: If primals lead, execute procedure **Initialize(Primals Lead)** and go to Step 2; else, duals lead, execute procedure **Initialize(Duals Lead)** and go to Step 3.

Step 2: Set $k \leftarrow k+1$ and $t \leftarrow 1$. Execute procedure **Solve(Primals Lead)** and go to Step 4.

Step 3: Set $k \leftarrow k+1$ and $t \leftarrow T$. Execute procedure **Solve(Duals Lead)** and go to Step 4.

Step 4: If $(z_{\text{UB}} - z_{\text{LB}}) \leq |z_{\text{LB}}|\,\varepsilon$, stop with ε-optimal procedure; else if $(z_{\text{UB}} - z_{\text{LB}}) \geq diff$, stop with diverging solution; else, set $diff \leftarrow (z_{\text{UB}} - z_{\text{LB}})$ and return to Step 2 if primals lead or to Step 3 if duals lead.

Parameter k is the cycle counter and parameters z_{LB} and z_{UB} record the best found lower and upper bounds respectively. Parameter $diff$ records the difference between bounds, $(z_{\text{UB}} - z_{\text{LB}})$, at the conclusion of the previous cycle. The four referenced procedures are detailed below.

B.2 Initialization Procedures

Procedure **Initialize(Duals Lead)** initializes the algorithm when duals lead by obtaining a solution to each nodal dual subproblem in the terminal period. Solutions to the dual subproblems in the terminal period are not required thereafter in the duals lead version. Procedure **Initialize(Primals Lead)** initializes the algorithm when primals lead by obtaining a solution to the nodal primal subproblem in the first period. Solutions to the primal subproblem at the first period node are not required thereafter in the primals lead version. Procedure **Initialize(Duals Lead)** is listed first followed by procedure **Initialize(Primals Lead)**. Each listing is followed by explanatory remarks.

Procedure Initialize(Duals Lead)

Step 1: Set $z_{\text{UB}}^{(1)} \leftarrow 0$ and $h \leftarrow 0$. Go to Step 2.

Step 2: If $h = H_T$, return to algorithm **MDPCA**. Solve the nodal dual subproblem [4.20] at node $(T, h) = [\bullet]_T$, let the optimal solution and objective function value be $\pi^{[\bullet]_T}$ and $\tilde{z}_{\text{UB}}$, set $z_{\text{UB}}^{(1)} \leftarrow z_{\text{UB}}^{(1)} + \tilde{z}_{\text{UB}}$ and repeat Step 2.

Parameter $z_{\text{UB}}^{(k)}$ records the upper bound for the current cycle, k, by accumulating the objective function values for the nodal dual subproblems. Parameter h indexes the nodes in the current period (recall that H_t is the total number of nodes in period t).

Procedure Initialize(Primals Lead)

Step 1: Solve the nodal primal subproblem [4.18] at node $(1, 1) = []$, let the optimal solution and objective function value be $\mathbf{x}^{[]}$ and $z_{\text{LB}}^{(1)}$ and return to algorithm **MDPCA**.

Parameter $z_{\text{LB}}^{(k)}$ serves the same purpose for the lower bound at the current cycle as does parameter $z_{\text{UB}}^{(k)}$ for the upper bound.

B.3 Solution Procedures

Procedure **Solve(Duals Lead)** executes one complete cycle of the duals lead version of the algorithm where each cycle starts with the nodal dual subproblems in period $T-1$. Information obtained at the conclusion of each lead dual half-cycle from the solution to the dual subproblem at the node in the first period is used to start the tail primal half-cycle in the second period. Solution information obtained at the conclusion of each primal half-cycle for the primal subproblems at the nodes in the terminal period is used to start the next cycle with the dual subproblems in period $T-1$. Procedure **Solve(Primals Lead)** executes one complete cycle of the primals lead version of the algorithm where each cycle starts with the nodal primal subproblems in the second period. Information obtained at the conclusion of each lead primal half-cycle from the solutions to the primal subproblems at the nodes in the terminal period is used to start the tail dual half-cycle in period $T-1$. Solution information obtained at the conclusion of each dual half-cycle for the dual subproblem at the first period node is used to start the next cycle with the primal subproblems in the second period. Procedure **Solve(Duals Lead)** is listed first followed by procedure **Solve(Primals Lead)**. Each listing is followed by explanatory remarks.

Procedure Solve(Duals Lead)

Step 1: If $t = 2$, go to Step 3; else, set $t \leftarrow t - 1$, $h \leftarrow 0$ and go to Step 2.

Step 2: If $h = H_t$, return to Step 1. Set $h \leftarrow h + 1$ and solve the nodal dual subproblem [4.21] at node $(t, h) = [\bullet]_t$. Let the optimal solution and objective function value be $\boldsymbol{\pi}^{[\bullet]_t}$ and $\tilde{z}_{\text{UB}}$. Set $z_{\text{UB}}^{(k)} \leftarrow z_{\text{UB}}^{(k)} + \tilde{z}_{\text{UB}}$ and repeat Step 2.

Step 3: Set $t \leftarrow 1$ and solve nodal dual subproblem [4.22] at node $(1, 1) = []$. Let optimal solution, dual multipliers, and objective function value be $\boldsymbol{\pi}^{[]}$, $\mathbf{x}^{[]}$, and $\tilde{z}_{\text{UB}}$. Set $z_{\text{UB}}^{(k)} \leftarrow z_{\text{UB}}^{(k)} + \tilde{z}_{\text{UB}}$, $z_{\text{LB}}^{(k)} \leftarrow \tilde{z}_{\text{UB}} + \mathbf{x}'^{[]} \sum_{l_1=1}^{L_1} \mathbf{B}'^{[\bullet]_2} \boldsymbol{\pi}^{[\bullet]_2}$ and go to Step 4.

Step 4: Set $z_{\text{UB}} = \max\left(z_{\text{UB}}, z_{\text{UB}}^{(k)}\right)$ and go to Step 5.

Step 5: If $t = T - 1$, go to Step 7; else, set $t \leftarrow t + 1$, $h \leftarrow 0$ and go to Step 6.

Step 6: If $h = H_t$, return to Step 5. Set $h \leftarrow h + 1$ and solve the nodal primal subproblem [4.19] at node $(t, h) = [\bullet]_t$. Let the optimal solution and objective function value be $\mathbf{x}^{[\bullet]_t}$ and $\tilde{z}_{\text{LB}}$. Set $z_{\text{LB}}^{(k)} \leftarrow z_{\text{LB}}^{(k)} + \tilde{z}_{\text{LB}}$ and repeat Step 6.

Step 7: Set $z_{\text{UB}}^{(k+1)} \leftarrow 0$, $h \leftarrow 0$ and go to Step 8.

Step 8: If $h = H_T$, go to Step 9. Set $h \leftarrow h + 1$ and solve the nodal primal subproblem [4.19] at node $(T, h) = [\bullet]_T$. Let the optimal solution, dual multipliers, and objective function value be $\mathbf{x}^{[\bullet]_T}$, $\boldsymbol{\pi}^{[\bullet]_T}$, and $\tilde{z}_{\text{LB}}$. Set $z_{\text{LB}}^{(k)} \leftarrow z_{\text{LB}}^{(k)} + \tilde{z}_{\text{LB}}$, $z_{\text{UB}}^{(k+1)} \leftarrow z_{\text{UB}}^{(k+1)} + \tilde{z}_{\text{LB}} + \boldsymbol{\pi}'^{[\bullet]_T} \mathbf{B}^{[\bullet]_T} \mathbf{x}^{[\bullet]_T}$ and repeat Step 8.

Step 9: Set $z_{\text{LB}} = \min\left(z_{\text{LB}}, z_{\text{LB}}^{(k)}\right)$ and return to algorithm **MDPCA**.

Appendix B Algorithm MDPCA Listings

Cumulative bounding values $z_{\text{LB}}^{(k)}$ and $z_{\text{UB}}^{(k+1)}$ are initialized in Steps 3 and 8 for the next primal and dual half-cycles respectively. The best upper and lower bounds found to date are updated in Steps 4 and 9 respectively.

Procedure Solve(Primals Lead)

Step 1: If $t = T - 1$, go to Step 3; else, set $t \leftarrow t + 1$, $h \leftarrow 0$ and go to Step 2.

Step 2: If $h = H_t$, return to Step 1. Set $h \leftarrow h + 1$ and solve the nodal primal subproblem [4.19] at node $(t, h) = [\bullet]_t$. Let the optimal solution and objective function value be $\mathbf{x}^{[\bullet]_t}$ and $\tilde{z}_{\text{LB}}$. Set $z_{\text{LB}}^{(k)} \leftarrow z_{\text{LB}}^{(k)} + \tilde{z}_{\text{LB}}$ and repeat Step 2.

Step 3: Set $z_{\text{UB}}^{(k)} \leftarrow 0$, $t \leftarrow T$, $h \leftarrow 0$ and go to Step 4.

Step 4: If $h = H_T$, go to Step 5. Set $h \leftarrow h + 1$ and solve the nodal primal subproblem [4.19] at node $(T, h) = [\bullet]_T$. Let the optimal solution, dual multipliers, and objective function value be $\mathbf{x}^{[\bullet]_T}$, $\boldsymbol{\pi}^{[\bullet]_T}$, and $\tilde{z}_{\text{LB}}$. Set $z_{\text{LB}}^{(k)} \leftarrow z_{\text{LB}}^{(k)} + \tilde{z}_{\text{LB}}$, $z_{\text{UB}}^{(k)} \leftarrow z_{\text{UB}}^{(k)} + \tilde{z}_{\text{LB}} + \boldsymbol{\pi}'^{[\bullet]_T}\mathbf{B}^{[\bullet]_T}\mathbf{x}^{[\bullet]_T}$ and repeat Step 4.

Step 5: Set $z_{\text{LB}} = \min\left(z_{\text{LB}}, z_{\text{LB}}^{(k)}\right)$ and go to Step 6.

Step 6: If $t = 2$, go to Step 8; else, set $t \leftarrow t - 1$, $h \leftarrow 0$ and go to Step 7.

Step 7: If $h = H_t$, return to Step 6. Set $h \leftarrow h + 1$ and solve the nodal dual subproblem [4.21] at node $(t, h) = [\bullet]_t$. Let the optimal solution and objective function value be $\boldsymbol{\pi}^{[\bullet]_t}$ and $\tilde{z}_{\text{UB}}$. Set $z_{\text{UB}}^{(k)} \leftarrow z_{\text{UB}}^{(k)} + \tilde{z}_{\text{UB}}$ and repeat Step 7.

Step 8: Solve nodal dual subproblem [4.22] at node $(1,1) = []$. Let optimal solution, dual multipliers, and objective function value be $\pi^{[]}$, $\mathbf{x}^{[]}$, and $\tilde{z}_{\text{UB}}$. Set $z_{\text{UB}}^{(k)} \leftarrow z_{\text{UB}}^{(k)} + \tilde{z}_{\text{UB}}$, $z_{\text{LB}}^{(k+1)} \leftarrow \tilde{z}_{\text{UB}} + \mathbf{x}'^{[]} \sum_{l_1=1}^{L_1} \mathbf{B}'^{[\bullet]_2} \pi^{[\bullet]_2}$ and go to Step 9.

Step 9: Set $z_{\text{UB}} = \max\left(z_{\text{UB}}, z_{\text{UB}}^{(k)}\right)$ and return to algorithm **MDPCA**.

Cumulative bounding values $z_{\text{UB}}^{(k)}$ and $z_{\text{LB}}^{(k+1)}$ are initialized in Steps 4 and 8 for the next dual and primal half-cycles respectively. The best lower and upper bounds found to date are updated in Steps 5 and 9 respectively.

Figure B.1 is the flow chart for algorithm **MDPCA**. Figure B.2 shows the flow-charts for both initialization procedures with procedure **Initialize(Duals Lead)** to the left of procedure **Initialize(Primals Lead)**. Flowcharts for procedures **Solve(Duals Lead)** and **Solve(Primals Lead)** are at Figures B.3 and B.4 respectively.

B.4 Algorithm Modifications

Modifications are easily made to algorithm **MDPCA** depending upon the desired results. The coded version of the algorithm includes several modifications. Options are available to force the algorithm to terminate (return to the calling routine) after the first complete cycle or after the first lead half-cycle (in either the duals lead or primals lead application). The algorithm will also terminate after a specified number of complete cycles regardless of the status of the other two termination criteria – ε-optimal solution or a diverging solution.

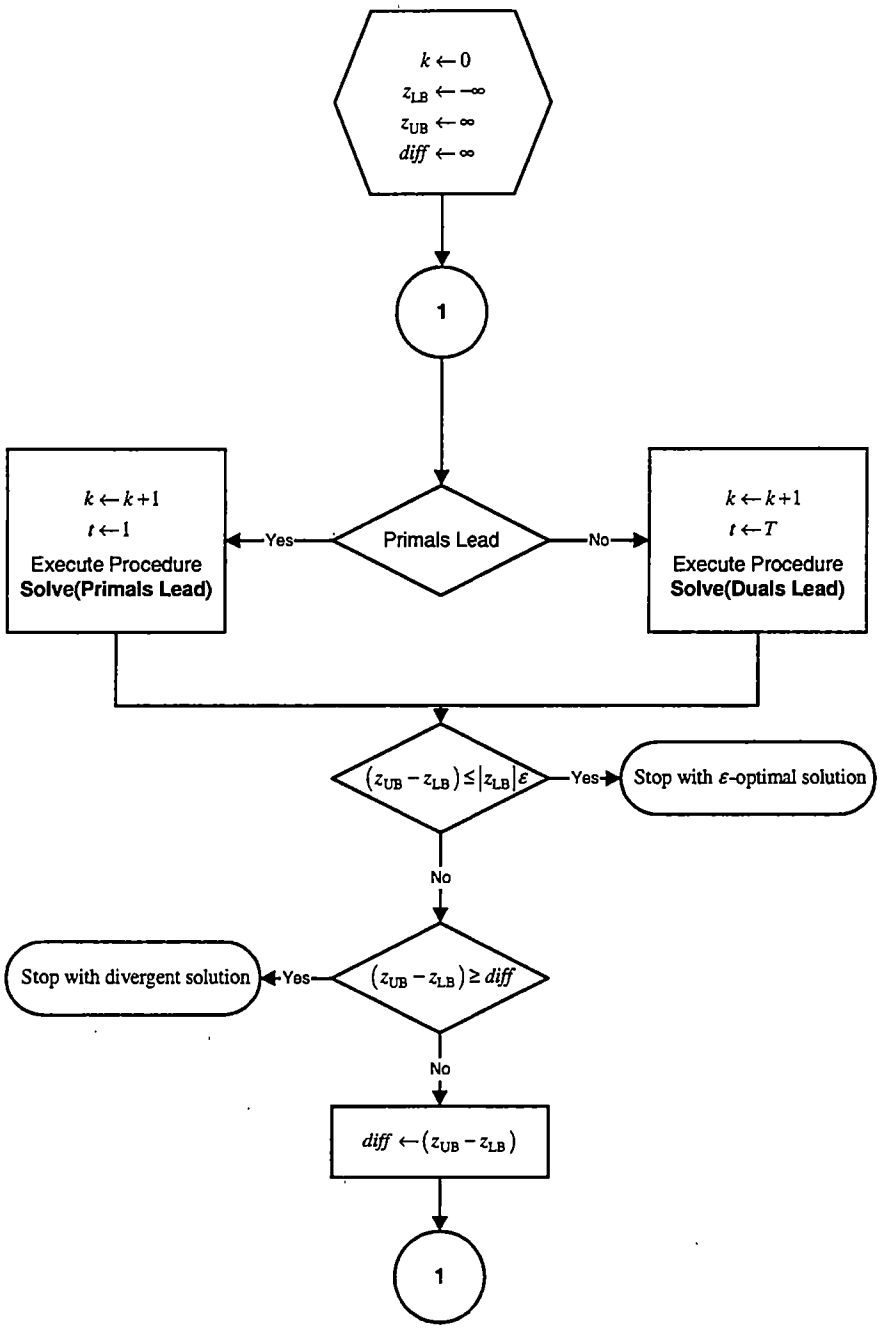
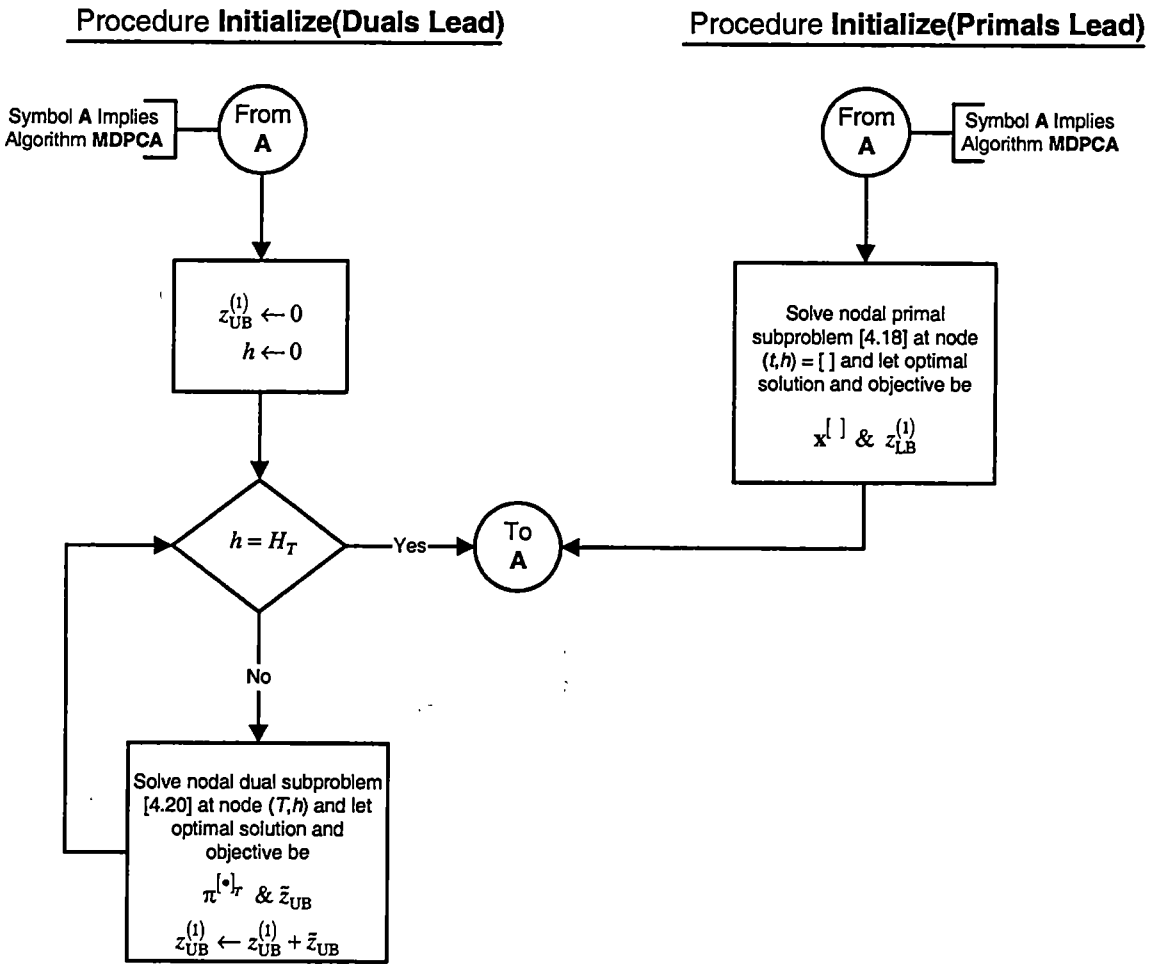
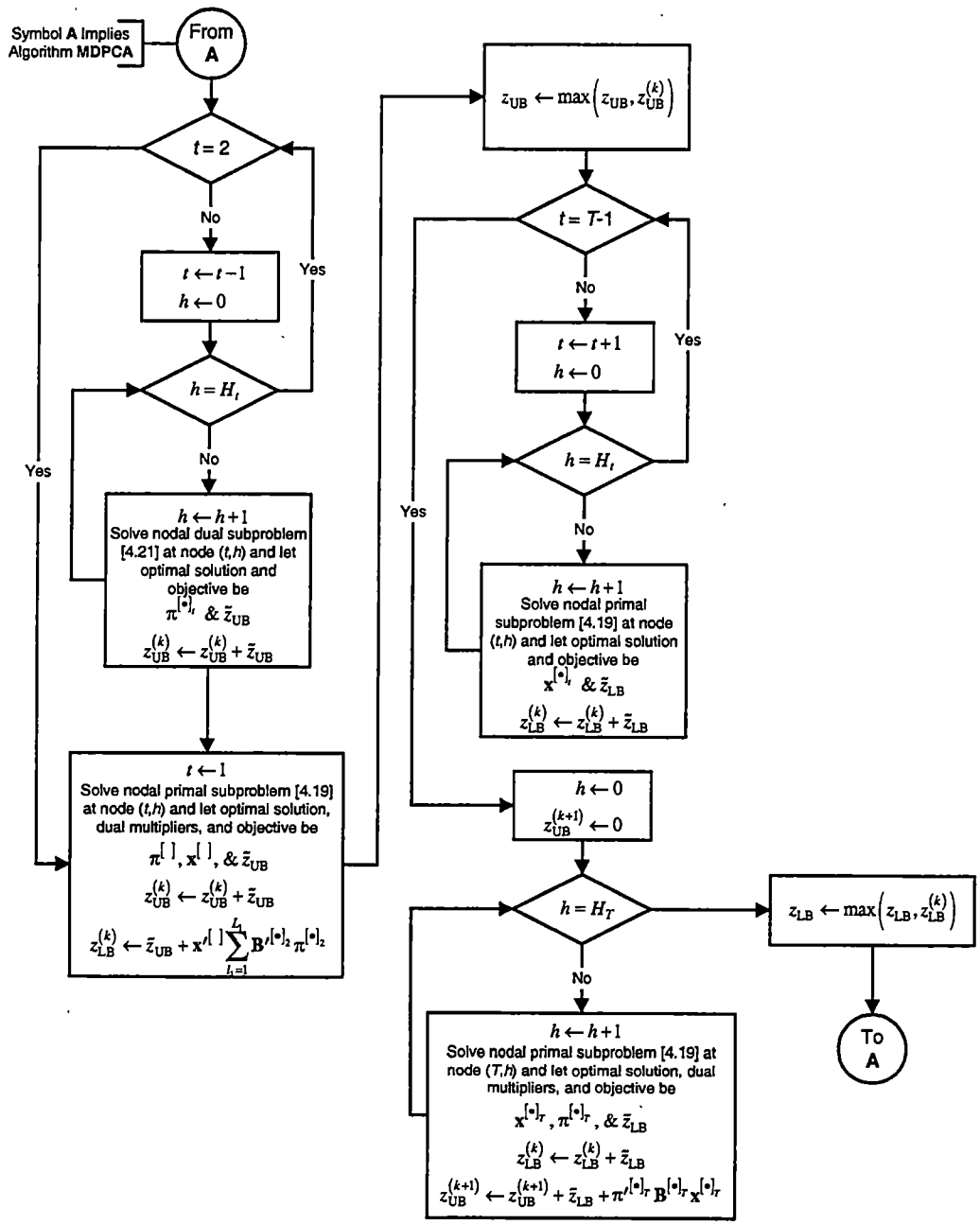


Figure B.1: Algorithm **MDPCA** Flowchart



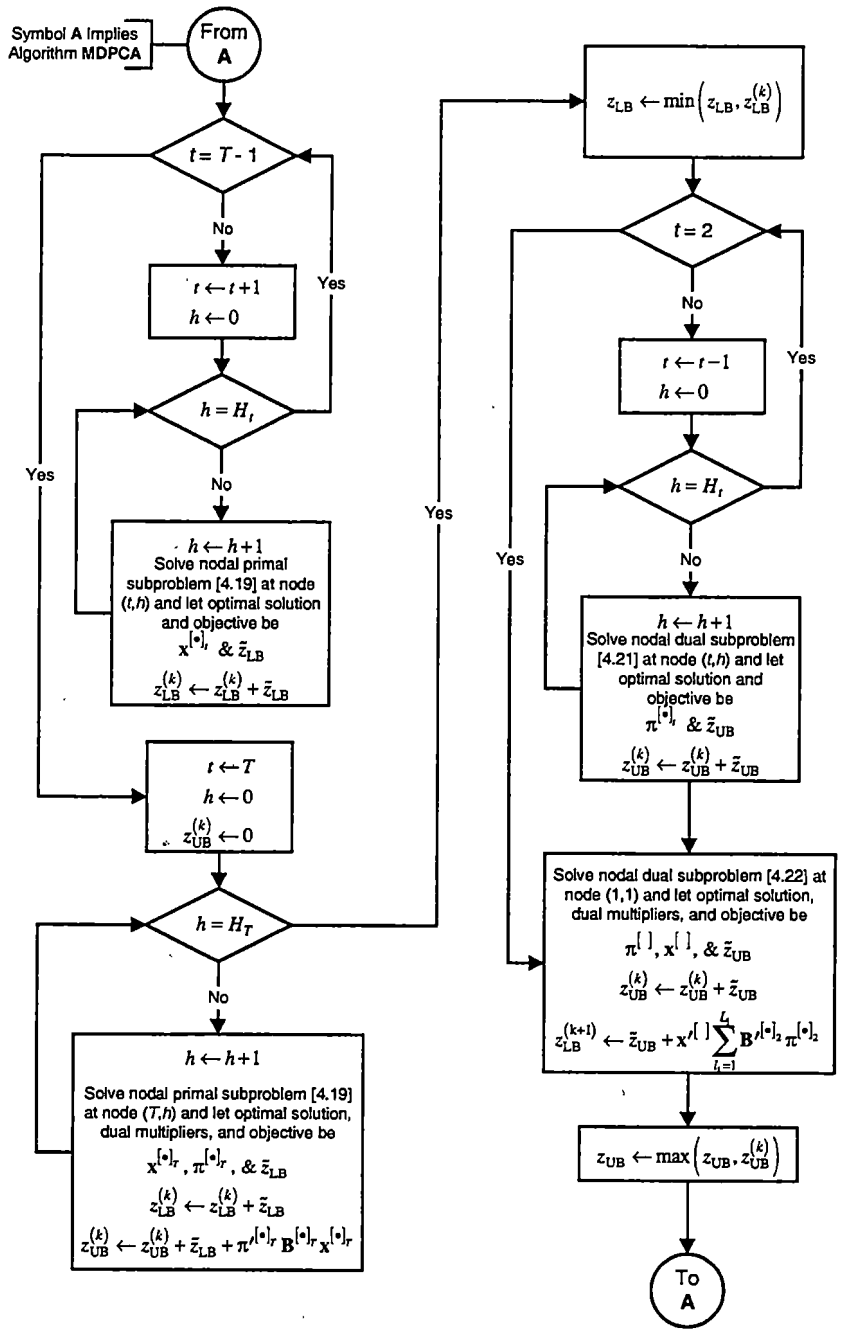
NOTE: Use equation [2.7] to determine path vector $[\bullet]_T$ given period-index label (T,h)!

Figure B.2: Procedures **Initialize(Duals Lead)** and **Initialize(Primals Lead)** Flowcharts



NOTE: Use equation [2.7] to determine path vector $[\bullet]_t$ given period-index label (t,h)!

Figure B.3: Procedure **Solve(Duals Lead)** Flowchart



NOTE: Use equation [2.7] to determine path vector $[\bullet]_t$ given period-index label (t,h)!

Figure B.4: Procedure **Solve(Primals Lead)** Flowchart

*Appendix B Algorithm **MDPCA** Listings*

Algorithm **MDPCA** can therefore be used to obtain an initial feasible solution to either the primal or the dual problem in a relatively efficient manner. These solutions can then be used to initiate either Dantzig-Wolfe or L-Shaped Decomposition as appropriate.

Appendix C
Piece-Wise Linear Approximations

Procedures for assigning slopes and break points for the piece-wise linear (PWL) approximations to the quadratic slippage function and the quadratic downside deviation function are developed in this appendix. All derivations are based upon the slippage function since a quadratic downside deviation function can be considered as a special case of the slippage function. General conditions assumed in the derivations are first established. Procedures for assigning slopes and break points are then developed for two criteria – controlling the maximum relative error in any PWL segment and minimizing the average absolute error over all PWL segments in a bounded region. Procedural descriptions are followed by discussions concerning applications of the procedures under general conditions and under several special conditions. The final section details using the procedures developed for the slippage function to determine PWL approximation parameters for quadratic downside deviation.

C.1 General Conditions

All PWL approximation derivations for slippage are developed for a generic security with index n where $1 \leq n \leq N$ and N is the number of securities represented in the model. Let $\hat{y}_n$ be the number of positions held in security n at the end of a period (or at the start of the planning horizon) and let $\tilde{y}_n$ represent the number of positions held in security n after

trading is completed at the start of the next period. Then, $x_n = |\tilde{y}_n - \hat{y}_n|$ is the absolute transaction amount for security n and the associated transaction cost, or slippage, is:

$$\widetilde{\mathfrak{S}}_n(x_n) = \begin{cases} x_n(a_{1n}x_n + \kappa_n), & 0 \le x_n < X_n^{\text{cut}}, \\ x_n\left[(a_{1n} + a_{2n})x_n + \kappa_n - a_{2n}X_n^{\text{cut}}\right], & X_n^{\text{cut}} \le x_n < \infty, \end{cases} \quad [C.1]$$

where the parameters are defined in Section 5.1.1. Let K_1 and K_2 be the numbers of desired PWL break points, Ψ_{kn}, $k = 1, \ldots, K = K_1 + K_2$, in slippage regions one, $0 \le x_n < X_n^{\text{cut}}$, and two, $X_n^{\text{cut}} \le x_n < \infty$, respectively. Derivations are developed assuming the following general conditions exist:

$$\begin{aligned} 1 \le K_1 < \infty, \quad & [C.2] \\ 1 \le K_2 < \infty, \\ 0 < X_n^{\text{cut}} < \infty, \\ 0 < a_{1n} < \infty, \\ -a_{1n} < a_{2n} < \infty. \end{aligned}$$

Furthermore, break points are to be placed at the origin, $\Psi_{0n} = 0$, and at the region boundary, $\Psi_{K_1 n} = X_n^{\text{cut}}$. Figure 5.2 on page 171 illustrates the PWL slippage approximations for $K_1 = K_2 = 2$.

Let ς_1 and ς_2 represent user specified maximum desired relative errors between PWL and actual slippage for all securities in the first and second slippage regions respectively. Define the triplet $[\tilde{a}_{1n}, \tilde{a}_{2n}, \tilde{\varsigma}]$ as:

$$[\tilde{a}_{1n}, \tilde{a}_{2n}, \tilde{\varsigma}] = \begin{cases} [a_{1n}, \kappa_n, \varsigma_1], & 0 \le x_n < X_n^{\text{cut}}, \\ \left[(a_{1n} + a_{2n}), (\kappa_n - a_{2n}X_n^{\text{cut}}), \varsigma_2\right], & X_n^{\text{cut}} \le x_n < \infty, \end{cases} \quad [C.3]$$

so that the slippage function can be represented by:

$$\widetilde{\mathfrak{S}}_n(x_n) = x_n(\tilde{a}_{1n}x_n + \tilde{a}_{2n}). \qquad [C.4]$$

Let the linear functions

$$\widehat{\mathfrak{S}}_{kn}(x_n) = \Phi_{kn}x_n + v_{kn},\ \Psi_{(k-1)n} \leq x_n < \Psi_{kn}, \qquad [C.5]$$

be the slippage approximating functions in PWL segments $k = 1, \ldots, K$. The slope, Φ_{kn}, in segment $k \in \mathsf{K} = \{1, \ldots, K\}$ is:

$$\begin{aligned}\Phi_{kn} &= \frac{\Psi_{kn}(\tilde{a}_{1n}\Psi_{kn} + \tilde{a}_{2n}) - \Psi_{(k-1)n}\left(\tilde{a}_{1n}\Psi_{(k-1)n} + \tilde{a}_{2n}\right)}{\Psi_{kn} - \Psi_{(k-1)n}} \\ &= \frac{\tilde{a}_{1n}\left(\Psi_{kn}^2 - \Psi_{(k-1)n}^2\right) + \tilde{a}_{2n}\left(\Psi_{kn} - \Psi_{(k-1)n}\right)}{\Psi_{kn} - \Psi_{(k-1)n}} \\ \Phi_{kn} &= \tilde{a}_{1n}\left(\Psi_{(k-1)n} + \Psi_{kn}\right) + \tilde{a}_{2n}, \qquad [C.6a]\end{aligned}$$

and the vertical axis intercept, v_{kn}, for segment $k \in \mathsf{K}$ is determined by setting $\widehat{\mathfrak{S}}_{kn}(\Psi_{kn}) = \widetilde{\mathfrak{S}}_n(\Psi_{kn})$:

$$\begin{aligned}v_{kn} &= \Psi_{kn}(\tilde{a}_{1n}\Psi_{kn} + \tilde{a}_{2n}) - \left[\tilde{a}_{1n}\left(\Psi_{kn} + \Psi_{(k-1)n}\right) + \tilde{a}_{2n}\right]\Psi_{kn} \\ &= \tilde{a}_{1n}\Psi_{kn}^2 + \tilde{a}_{2n}\Psi_{kn} - \tilde{a}_{1n}\Psi_{kn}^2 - \tilde{a}_{1n}\Psi_{(k-1)n}\Psi_{kn} - \tilde{a}_{2n}\Psi_{kn} \\ v_{kn} &= -\tilde{a}_{1n}\Psi_{(k-1)n}\Psi_{kn}. \qquad [C.6b]\end{aligned}$$

Note that each term on the right-hand-side of equation [C.6b] is nonnegative implying that:

$$v_{kn} \leq 0,\ k = 1, \ldots, K. \qquad ([C.6c])$$

Finally, define the absolute error, $\mathfrak{R}_n(x_n)$, between the PWL approximate slippage and the actual slippage at some x_n as:

$$\mathfrak{R}_n(x_n) = \widehat{\mathfrak{S}}_{kn}(x_n) - \widetilde{\mathfrak{S}}_n(x_n), \Psi_{(k-1)n} \leq x_n < \Psi_{kn}. \qquad [C.7]$$

Note that the actual slippage function, $\widetilde{\mathfrak{S}}_n(x_n)$ as defined by equation [C.4], is convex in each slippage region since $\tilde{a}_{1n} > 0$ in both slippage regions under the general conditions [C.2] ($\widetilde{\mathfrak{S}}_n(x_n)$ has a positive second derivative). Therefore, $\mathfrak{R}_n\left(\Psi_{(k-1)n}\right) = \mathfrak{R}_n(\Psi_{kn}) = 0$ and $\mathfrak{R}_n(x_n) > 0$ for $\Psi_{(k-1)n} < x_n < \Psi_{kn}$.

The objective of the PWL approximation scheme is to assign break points $0 = \Psi_{0n} < \Psi_{1n} < \cdots < \Psi_{kn} < \cdots < \Psi_{Kn} = \infty$ such that the slope in each PWL segment is strictly smaller than the slope in the next segment: $\Phi_{1n} < \cdots < \Phi_{kn} < \cdots < \Phi_{Kn}$. Slopes within the same slippage region will naturally increase in value from segment to segment due to the convexity of $\widetilde{\mathfrak{S}}_n(x_n)$ in each region. Special attention, however, must be given to the assignment of the first break point, $\Psi_{(K_1+1)n}$, in the second slippage region when the slope adjustment factor is negative, i.e., $a_{2n} < 0$. The actual slippage function is not convex across the region boundary, X_n^{cut}, when $a_{2n} < 0$ and care must be taken to insure that $\Phi_{(K_1+1)n} > \Phi_{K_1 n}$. Break points are assigned in order to control the approximation error in either one of two ways as described in the next two sections.

C.2 Maximum Relative Error Procedure

The goal of the maximum relative error procedure is to assign break points so that no relative error is greater than some specified positive value. This goal can be achieved by insuring that the maximum relative error in each PWL segment is no larger than the specified value. The procedure may be used in both slippage regions with Ψ_{0n} and $\Psi_{K_1 n}$ preassigned the values 0 and X_n^{cut} respectively.

Define the relative error between the approximate and actual slippage at some transaction amount x_n as

$$\mathcal{R}_n(x_n) = \frac{\mathfrak{R}_n(x_n)}{\widetilde{\mathfrak{S}}_n(x_n)},$$

where $\widetilde{\mathfrak{S}}_n(x_n)$ and $\mathfrak{R}_n(x_n)$ are defined by equations [C.4] and [C.7] respectively. Equations [C.3] through [C.7] imply that:

$$\begin{aligned}
\mathcal{R}_n(x_n) &= \frac{\widehat{\mathfrak{S}}_{kn}(x_n) - \widetilde{\mathfrak{S}}_n(x_n)}{\widetilde{\mathfrak{S}}_n(x_n)} \\
&= \frac{\Phi_{kn} x_n + v_{kn} - \tilde{a}_{1n} x_n^2 - \tilde{a}_{2n} x_n}{\tilde{a}_{1n} x_n^2 + \tilde{a}_{2n} x_n} \\
\mathcal{R}_n(x_n) &= -\frac{\tilde{a}_{1n} x_n^2 - (\Phi_{kn} - \tilde{a}_{2n}) x_n - v_{kn}}{\tilde{a}_{1n} x_n^2 + \tilde{a}_{2n} x_n}. \qquad \text{[C.8]}
\end{aligned}$$

The first derivative of the relative error function with respect to x_n is then:

$$\begin{aligned}
\frac{\partial \mathcal{R}_n(x_n)}{\partial x_n} &= -\frac{(\tilde{a}_{1n}x_n^2 + \tilde{a}_{2n}x_n)\left[2\tilde{a}_{1n}x_n - (\Phi_{kn} - \tilde{a}_{2n})\right]}{(\tilde{a}_{1n}x_n^2 + \tilde{a}_{2n}x_n)^2} \\
&\quad + \frac{\left[\tilde{a}_{1n}x_n^2 - (\Phi_{kn} - \tilde{a}_{2n})\,x_n - v_{kn}\right](2\tilde{a}_{1n}x_n + \tilde{a}_{2n})}{(\tilde{a}_{1n}x_n^2 + \tilde{a}_{2n}x_n)^2} \\
&= \frac{-2\tilde{a}_{1n}x_n^3 + \tilde{a}_{1n}(\Phi_{kn} - 3\tilde{a}_{2n})\,x_n^2 + \tilde{a}_{2n}(\Phi_{kn} - \tilde{a}_{2n})\,x_n}{(\tilde{a}_{1n}x_n^2 + \tilde{a}_{2n}x_n)^2} \\
&\quad + \frac{2\tilde{a}_{1n}x_n^3 - \tilde{a}_{1n}(2\Phi_{kn} - 3\tilde{a}_{2n})\,x_n^2 - \left[\tilde{a}_{2n}(\Phi_{kn} - \tilde{a}_{2n}) + 2\tilde{a}_{1n}v_{kn}\right]x_n - \tilde{a}_{2n}v_{kn}}{(\tilde{a}_{1n}x_n^2 + \tilde{a}_{2n}x_n)^2} \\
\frac{\partial \mathcal{R}_n(x_n)}{\partial x_n} &= -\frac{\tilde{a}_{1n}\Phi_{kn}x_n^2 + 2\tilde{a}_{1n}v_{kn}x_n + \tilde{a}_{2n}v_{kn}}{(\tilde{a}_{1n}x_n^2 + \tilde{a}_{2n}x_n)^2}. \qquad \text{[C.9]}
\end{aligned}$$

The maximum relative error in PWL segment $k \in \mathrm{K}$ will occur at $x_n = \hat{x}_n$ where the first derivative above is zero implying that:

$$\begin{aligned}
\hat{x}_n &= \frac{-2\tilde{a}_{1n}v_{kn} \pm \sqrt{4\tilde{a}_{1n}^2v_{kn}^2 - 4\tilde{a}_{1n}\tilde{a}_{2n}v_{kn}\Phi_{kn}}}{2\tilde{a}_{1n}\Phi_{kn}} \\
&= \frac{-2\tilde{a}_{1n}v_{kn} \pm 2\tilde{a}_{1n}v_{kn}\sqrt{1 - \frac{\tilde{a}_{2n}\Phi_{kn}}{\tilde{a}_{1n}v_{kn}}}}{2\tilde{a}_{1n}\Phi_{kn}} \\
\hat{x}_n &= -\frac{v_{kn}}{\Phi_{kn}}\left(1 \mp \sqrt{1 - \frac{\tilde{a}_{2n}\Phi_{kn}}{\tilde{a}_{1n}v_{kn}}}\right). \qquad \text{[C.10]}
\end{aligned}$$

Existence of at least one $\hat{x}_n$ satisfying equation [C.10] such that $\Psi_{(k-1)n} < \hat{x}_n < \Psi_{kn}$, $2 \leq k \leq K$, can be verified by the following theorem (e.g., see Taylor and Mann [193, page 27]):

Rolle's Theorem *Let $\mathfrak{F}(x)$ be a function which is continuous at each point of the closed interval $a \leq x \leq b$, has a derivative at each point in the open interval $a < x < b$, and is*

zero at the bounds: $\mathfrak{F}(a) = \mathfrak{F}(b) = 0$. *Then at least one* x, $a < x < b$, *exists such that*:

$$\frac{\mathrm{d}\mathfrak{F}(x)}{\mathrm{d}x} = 0.$$

Since $\mathcal{R}_n(x_n)$ satisfies the conditions of Rolle's Theorem for $k = 2, \ldots, K$, equation [C.10] must yield at least one value for $\hat{x}_n$ such that $\Psi_{(k-1)n} < \hat{x}_n < \Psi_{kn}$ with $\frac{\partial \mathcal{R}_n(x_n)}{\partial x_n} = 0$. Furthermore, there is only one such value for each $k \in \{2, \ldots, K\}$ and it is obtained by adding the radical value in the enclosed term of equation [C.10]. The latter statement is justified by noting that $\hat{x}_n \leq \Psi_{(k-1)n}$ when the radical value in the enclosed term in equation [C.10] is subtracted from one. If the enclosed term is negative, then $\hat{x}_n \leq 0 \leq \Psi_{(k-1)n}$ by inequality [C.6c] and construction of the break points. Then consider the case when the enclosed term of equation [C.10] is nonnegative and less than one after subtracting the radical value from one implying that

$$\hat{x}_n \leq -\frac{v_{kn}}{\Phi_{kn}},$$

which along with the condition $\hat{x}_n > \Psi_{(k-1)n}$ requires that:

$$\begin{aligned}
-\tfrac{v_{kn}}{\Phi_{kn}} > \; & \Psi_{(k-1)n}, \\
\Longrightarrow \quad & -v_{kn} > \Phi_{kn}\Psi_{(k-1)n}, \text{ since slope } \Phi_{kn} > 0 \text{ by construction}, \\
\Longrightarrow \quad & \Phi_{kn}\Psi_{(k-1)n} + v_{kn} < 0, \\
\Longrightarrow \quad & \widehat{\mathfrak{S}}_{kn}\left(\Psi_{(k-1)n}\right) = \widetilde{\mathfrak{S}}_n\left(\Psi_{(k-1)n}\right) < 0, \text{ by construction and equation [C.5]}.
\end{aligned}$$

The last expression cannot be valid since $\widetilde{\mathfrak{S}}_n\left(\Psi_{(k-1)n}\right) \geq 0$ for all $k = 1, \ldots, K$. Therefore, the unique value for $\hat{x}_n$ such that $\Psi_{(k-1)n} < \hat{x}_n < \Psi_{kn}$ is:

$$\hat{x}_n = -\frac{v_{kn}}{\Phi_{kn}}\left(1 + \sqrt{1 - \frac{\tilde{a}_{2n}\Phi_{kn}}{\tilde{a}_{1n}v_{kn}}}\right), 2 \leq k \leq K. \qquad \text{[C.11]}$$

Moreover, $\mathcal{R}_n(\hat{x}_n)$ must be a maximum value for the function in the segment $\Psi_{(k-1)n} \leq x_n \leq \Psi_{kn}$ since $\mathcal{R}_n\left(\Psi_{(k-1)n}\right) = \mathcal{R}_n(\Psi_{kn}) = 0$ and $\mathcal{R}_n(x_n) > 0$ for all x_n such that $\Psi_{(k-1)n} < x_n < \Psi_{kn}$. Given $\Psi_{(k-1)n}$, $k = 2, \ldots, K-1$, the maximum relative error can then be controlled by assigning a value to Ψ_{kn} that insures that

$$\mathcal{R}_n(\hat{x}_n) \leq \tilde{\varsigma},$$

where $\tilde{\varsigma}$ is the maximum desired relative error as defined by equation [C.3].

Equation [C.11] does not apply for the first PWL segment at $k = 1$ since the relative error at the origin is undefined:

$$\mathcal{R}_n(0) = \frac{\mathfrak{R}_n(0)}{\widetilde{\mathfrak{S}}_n(0)} = \frac{\widehat{\mathfrak{S}}_{1n}(0) - \widetilde{\mathfrak{S}}_n(0)}{\widetilde{\mathfrak{S}}_n(0)} = \frac{0}{0}.$$

Hence, the conditions of Rolle's Theorem are not satisfied at $x_n = 0$ and equation [C.11] cannot be used to set a value for Ψ_{1n}. The procedure to determine Ψ_{1n} is described at the conclusion of this section.

Assume that $\Psi_{jn} = \widetilde{\Psi}_{jn}$, $j = 1, \ldots, k-1$, $2 \leq k \leq K$, have been fixed so that Ψ_{kn} is the next break point to be determined such that:

$$\mathcal{R}_n(\hat{x}_n) = \tilde{\varsigma}, \widetilde{\Psi}_{(k-1)n} < \hat{x}_n < \Psi_{kn}.$$

Equation [C.8] defining $\mathcal{R}_n(x_n)$ then implies that:

$$\begin{aligned} -\tilde{a}_{1n}\hat{x}_n^2 + (\Phi_{kn} - \tilde{a}_{2n})\,\hat{x}_n + v_{kn} = &\quad (\tilde{a}_{1n}\hat{x}_n^2 + \tilde{a}_{2n}\hat{x}_n)\,\tilde{\varsigma}, \\ \Longrightarrow (1+\tilde{\varsigma})\,\tilde{a}_{1n}\hat{x}_n^2 + [(1+\tilde{\varsigma})\,\tilde{a}_{2n} - \Phi_{kn}]\,\hat{x}_n - v_{kn} = &\quad 0. \end{aligned}$$

The next break point, Ψ_{kn}, is the root of the above equation and can be determined with a Newton-Raphson/bisection procedure. Define $\mathfrak{F}_{kn}(\Psi_{kn})$, $k = 2, \ldots, K$, as:

$$\mathfrak{F}_{kn}(\Psi_{kn}) = (1+\tilde{\varsigma})\,\tilde{a}_{1n}\hat{x}_n^2 + [(1+\tilde{\varsigma})\,\tilde{a}_{2n} - \Phi_{kn}]\,\hat{x}_n - v_{kn} \qquad \text{[C.12]}$$

where,

$$\hat{x}_n = -\frac{v_{kn}}{\Phi_{kn}}\left(1 + \sqrt{1 - \frac{\tilde{a}_{2n}\Phi_{kn}}{\tilde{a}_{1n}v_{kn}}}\right), \qquad \text{[C.13a]}$$

$$\Phi_{kn} = \tilde{a}_{1n}\left(\Psi_{kn} + \widetilde{\Psi}_{(k-1)n}\right) + \tilde{a}_{2n}, \qquad \text{[C.13b]}$$

$$v_{kn} = -\tilde{a}_{1n}\widetilde{\Psi}_{(k-1)n}\Psi_{kn}. \qquad \text{[C.13c]}$$

Then, $\Psi_{kn} = \widetilde{\Psi}_{kn}$ is to be determined such that $\mathfrak{F}_{kn}\left(\widetilde{\Psi}_{kn}\right) = 0$. Note that equations [C.13a-c] are convenient reproductions of equations [C.11] and [C.6a-b] respectively. Each of the left-hand-side terms in these three equations is treated as a function of Ψ_{kn} although functional notation has been omitted for simplicity. The first derivatives of the functions

defined by equations [C.12]-[C.13c] with respect to Ψ_{kn} are:

$$\frac{\partial \mathfrak{F}_{kn}\left(\Psi_{kn}\right)}{\partial \Psi_{kn}} = \left[\left(1+\tilde{\varsigma}\right)\left(2\tilde{a}_{1n}\hat{x}_n + \tilde{a}_{2n}\right) - \Phi_{kn}\right]\frac{\partial \hat{x}_n}{\partial \Psi_{kn}} - \hat{x}_n\frac{\partial \Phi_{kn}}{\partial \Psi_{kn}} - \frac{\partial v_{kn}}{\partial \Psi_{kn}}, \qquad [C.14]$$

$$\frac{\partial \hat{x}_n}{\partial \Psi_{kn}} = \left[\frac{\tilde{a}_{2n}}{2\tilde{a}_{1n}\left(\Phi_{kn}\hat{x}_n + v_{kn}\right)} + \frac{\hat{x}_n}{v_{kn}}\right]\left(\frac{\Phi_{kn}\frac{\partial v_{kn}}{\partial \Psi_{kn}} - v_{kn}\frac{\partial \Phi_{kn}}{\partial \Psi_{kn}}}{\Phi_{kn}}\right), \qquad [C.15a]$$

$$\frac{\partial \Phi_{kn}}{\partial \Psi_{kn}} = \tilde{a}_{1n}, \qquad [C.15b]$$

$$\frac{\partial v_{kn}}{\partial \Psi_{kn}} = -\tilde{a}_{1n}\widetilde{\Psi}_{(k-1)n}. \qquad [C.15c]$$

The next value, $\Psi_{kn}^{(G)}$, for the break point in question at iteration G of the Newton-Raphson procedure is

$$\Psi_{kn}^{(G)} = \Psi_{kn}^{(G-1)} - \frac{\mathfrak{F}_{kn}\left(\Psi_{kn}^{(G-1)}\right)}{\left.\frac{\partial \mathfrak{F}_{kn}(\Psi_{kn})}{\partial \Psi_{kn}}\right|_{\Psi_{kn}=\Psi_{kn}^{(G-1)}}},$$

where equations [C.12] through [C.15c] are used to evaluate the necessary terms. A switch is made to the bisection procedure if the Newton-Raphson procedure begins to diverge. The search terminates with $\widetilde{\Psi}_{kn} = \Psi_{kn}^{(G)}$ when $\left|\mathfrak{F}_{kn}\left(\Psi_{kn}^{(G)}\right)\right| < \varepsilon$ for some specified tolerance $\varepsilon > 0$.

The maximum relative error procedure starts by finding the upper break point, $\widetilde{\Psi}_{1n}$, for the first PWL segment. The Newton-Raphson/bisection procedure cannot be applied in this segment since the relative error as defined by equation [C.8] is undefined at the origin. However, since $\widetilde{\Psi}_{0n} = 0$, equation [C.13c] indicates that $v_{1k} = 0$ so that the first segment relative error may be redefined as:

$$\widehat{\mathcal{R}}_n\left(x_n\right) = -\frac{\tilde{a}_{1n}x_n - \left(\Phi_{kn} - \tilde{a}_{2n}\right)}{\tilde{a}_{1n}x_n + \tilde{a}_{2n}}, 0 \le x_n < \Psi_{1n},$$

with first derivative:

$$\frac{\partial \widehat{\mathcal{R}}_n(x_n)}{\partial x_n} = -\frac{\tilde{a}_{1n}\Phi_{kn}}{(\tilde{a}_{1n}x_n + \tilde{a}_{2n})^2}.$$

This derivative is negative for all $x_n \geq 0$ implying that $\widehat{\mathcal{R}}_n(x_n)$ decreases monotonically with x_n. Therefore, the maximum relative error in the first PWL segment occurs at the origin with:

$$\begin{aligned}\widehat{\mathcal{R}}_n(0) &= \frac{\Phi_{kn} - \tilde{a}_{2n}}{\tilde{a}_{2n}} \\ &= \frac{\tilde{a}_{1n}(0 + \Psi_{1n}) + \tilde{a}_{2n} - \tilde{a}_{2n}}{\tilde{a}_{2n}} \\ \widehat{\mathcal{R}}_n(0) &= \frac{a_{1n}\Psi_{1n}}{\kappa_n}, \end{aligned} \qquad \text{[C.16]}$$

using equations [C.3] and [C.6a]. This result suggests the following procedure to determine $\widetilde{\Psi}_{1n}$:

1. define $\widehat{\Psi}_{1n}$ as:

$$\widehat{\Psi}_{1n} = \frac{\kappa_n}{a_{1n}}\varsigma_1, \qquad \text{[C.17]}$$

2. if $\widehat{\Psi}_{1n} < X_n^{\text{cut}}$, set $\widetilde{\Psi}_{1n} = \widehat{\Psi}_{1n}$ and use the Newton-Raphson/bisection procedure for $\widetilde{\Psi}_{kn}$, $k = 2, \ldots, K$; else, set $\widetilde{\Psi}_{1n} = X_n^{\text{cut}}$ and use the Newton-Raphson/bisection procedure for the break points in the second slippage region.

Note that there will be allocated but unassigned break points if $\widehat{\Psi}_{1n} \geq X_n^{\text{cut}}$ and $K_1 > 1$. Any excess break points may be assigned to the second slippage region or the associated memory can be returned to the system. On the other hand, there is no guarantee that the final break point in the first slippage region will coincide with the region boundary when $\widehat{\Psi}_{1n} < X_n^{\text{cut}}$. Any unassigned break points may be treated as above if the Newton-

Raphson/bisection procedure determines $\widetilde{\Psi}_{jn} \geq X_n^{\text{cut}}$ for some $j < K_1$. The Newton-Raphson/bisection procedure cannot be used in the second slippage region, however, if $\widetilde{\Psi}_{K_1 n} < X_n^{\text{cut}}$. This latter condition occurs if the number of break points, K_1, assigned to the first slippage region is incompatible with the specified maximum desired relative error, ς_1, for this region. Any break points already assigned in the first slippage region are deleted and all first region break points are assigned with the minimum average absolute error procedure described below prior to assigning break points in the second slippage region.

C.3 Minimum Average Absolute Error Procedure

The maximum relative error procedure is used to assign a finite number of PWL break points over either a finite or an infinite domain. The minimum average absolute error procedure, on the other hand, is applicable only over a finite domain. Therefore, the latter procedure may be used only in the first slippage region. The minimum average absolute error algorithm assigns the K_1 break points based upon two criteria:

1. break points must be assigned to the origin and to the region boundary, i.e., $\Psi_{0n} = 0$ and $\Psi_{K_1 n} = X_n^{\text{cut}}$, and
2. any remaining break points are assigned such that the average absolute error over the first slippage region is minimized.

Appendix C Piece-Wise Linear Approximations

Average absolute error between PWL and actual slippage is evaluated using the definition for the average of a function over a closed region. Given an integrable function $\mathfrak{F}(x)$ over a closed region $[a, b]$, $b > a$, the average function value, $\overline{\mathfrak{F}}([a, b])$, over this region is defined to be (e.g., see Taylor and Mann [193, pages 45-46]):

$$\overline{\mathfrak{F}}([a, b]) = \frac{1}{b-a} \int_a^b \mathfrak{F}(x)\, dx.$$

Let $\mathfrak{A}(\Psi_{\bullet n})$ represent the average absolute error between PWL and actual slippage in the first slippage region given the vector of first region break points,

$$\Psi_{\bullet n} = (\Psi_{0n}, \ldots, \Psi_{kn}, \ldots, \Psi_{K_1 n})'.$$

Criterion two above then requires that break points Ψ_{kn}, $k = 1, \ldots, K_1 - 1$, be assigned such that

$$\mathfrak{A}(\Psi_{\bullet n}) = \frac{1}{X_n^{\text{cut}}} \int_0^{X_n^{\text{cut}}} \mathfrak{R}_n(x_n)\, dx_n$$

be minimized. Equation [C.7] defining the absolute error, $\mathfrak{R}_n(x_n)$, implies that

$$\mathfrak{A}(\Psi_{\bullet n}) = \frac{1}{X_n^{\text{cut}}} \sum_{k=1}^{K_1} \left[\int_{\Psi_{(k-1)n}}^{\Psi_{kn}} \left[\widehat{\mathfrak{S}}_{kn}(x_n) - \widetilde{\mathfrak{S}}_n(x_n) \right] dx_n \right],$$

and equations [C.1] and [C.5] indicate that

$$\mathfrak{A}(\Psi_{\bullet n}) = \frac{1}{X_n^{\text{cut}}} \sum_{k=1}^{K_1} \left[\int_{\Psi_{(k-1)n}}^{\Psi_{kn}} \left[\Phi_{kn} x_n + v_{kn} - x_n (a_{1n} x_n + \kappa_n) \right] dx_n \right].$$

Applying equations [C.6a-b] and [C.3] to the integrand results in

$$\begin{aligned}\Phi_{kn}x_n + v_{kn} - x_n\left(a_{1n}x_n + \kappa_n\right) &= \left[a_{1n}\left(\Psi_{(k-1)n} + \Psi_{kn}\right) + \kappa_n\right]x_n - a_{1n}\Psi_{(k-1)n}\Psi_{kn} \\ &\quad -x_n\left(a_{1n}x_n + \kappa_n\right) \\ &= a_{1n}\left(\Psi_{(k-1)n} + \Psi_{kn}\right)x_n + \kappa_n x_n - a_{1n}\Psi_{(k-1)n}\Psi_{kn} \\ &\quad -a_{1n}x_n^2 - \kappa_n x_n \\ &= a_{1n}\left[\left(\Psi_{(k-1)n} + \Psi_{kn}\right)x_n - x_n^2 - \Psi_{(k-1)n}\Psi_{kn}\right],\end{aligned}$$

so that the average error function simplifies to

$$\mathfrak{A}\left(\Psi_{\bullet n}\right) = \frac{a_{1n}}{X_n^{\text{cut}}}\sum_{k=1}^{K_1}\left[\int_{\Psi_{(k-1)n}}^{\Psi_{kn}}\left[\left(\Psi_{(k-1)n} + \Psi_{kn}\right)x_n - x_n^2 - \Psi_{(k-1)n}\Psi_{kn}\right]\mathrm{d}x_n\right].$$

Performing the indicated integrations yields

$$\begin{aligned}\mathfrak{A}\left(\Psi_{\bullet n}\right) &= \frac{a_{1n}}{X_n^{\text{cut}}}\sum_{k=1}^{K_1}\left[\left(\frac{\Psi_{(k-1)n} + \Psi_{kn}}{2}x_n^2 - \frac{1}{3}x_n^3 - \Psi_{(k-1)n}\Psi_{kn}x_n\right)\Bigg|_{\Psi_{(k-1)n}}^{\Psi_{kn}}\right] \\ &= \frac{a_{1n}}{6X_n^{\text{cut}}}\sum_{k=1}^{K_1}\left[3\left(\Psi_{(k-1)n} + \Psi_{kn}\right)\left(\Psi_{kn}^2 - \Psi_{(k-1)n}^2\right) - 2\left(\Psi_{kn}^3 - \Psi_{(k-1)n}^3\right)\right. \\ &\qquad \left. -6\Psi_{(k-1)n}\Psi_{kn}\left(\Psi_{kn} - \Psi_{(k-1)n}\right)\right] \\ &= \frac{a_{1n}}{6X_n^{\text{cut}}}\sum_{k=1}^{K_1}\left[\left(\Psi_{kn} - \Psi_{(k-1)n}\right)\left(\Psi_{kn}^2 - 2\Psi_{(k-1)n}\Psi_{kn} - \Psi_{(k-1)n}^2\right)\right] \\ \mathfrak{A}\left(\Psi_{\bullet n}\right) &= \frac{a_{1n}}{6X_n^{\text{cut}}}\sum_{k=1}^{K_1}\left(\Psi_{kn} - \Psi_{(k-1)n}\right)^3. \qquad \text{[C.18]}\end{aligned}$$

The first partial derivatives of $\mathfrak{A}\left(\Psi_{\bullet n}\right)$ with respect to Ψ_{kn} for $k = 1, \ldots, K_1 - 1$ (recall that Ψ_{0n} and $\Psi_{K_1 n}$ are fixed) are then:

$$\begin{aligned}
\frac{\partial \mathfrak{A}\left(\Psi_{\bullet n}\right)}{\partial \Psi_{kn}} &= \frac{a_{1n}}{6X_n^{\text{cut}}} \frac{\partial}{\partial \Psi_{kn}} \left[\left(\Psi_{kn} - \Psi_{(k-1)n}\right)^3 + \left(\Psi_{(k+1)n} - \Psi_{kn}\right)^3 \right] \\
&= \frac{a_{1n}}{6X_n^{\text{cut}}} \left[3\left(\Psi_{kn} - \Psi_{(k-1)n}\right)^2 - 3\left(\Psi_{(k+1)n} - \Psi_{kn}\right)^2 \right] \\
&= \frac{a_{1n}}{2X_n^{\text{cut}}} \left[\Psi_{kn}^2 - 2\Psi_{(k-1)n}\Psi_{kn} + \Psi_{(k-1)n}^2 - \Psi_{(k+1)n}^2 + 2\Psi_{kn}\Psi_{(k+1)n} - \Psi_{kn}^2 \right] \\
&= \frac{a_{1n}}{2X_n^{\text{cut}}} \left[2\left(\Psi_{(k+1)n} - \Psi_{(k-1)n}\right)\Psi_{kn} - \left(\Psi_{(k+1)n}^2 - \Psi_{(k-1)n}^2\right) \right] \\
\frac{\partial \mathfrak{A}\left(\Psi_{\bullet n}\right)}{\partial \Psi_{kn}} &= \frac{a_{1n}}{2X_n^{\text{cut}}} \left(\Psi_{(k+1)n} - \Psi_{(k-1)n}\right) \left[2\Psi_{kn} - \left(\Psi_{(k-1)n} + \Psi_{(k+1)n}\right) \right]. \qquad \text{[C.19]}
\end{aligned}$$

Therefore,

$$\frac{\partial \mathfrak{A}\left(\Psi_{\bullet n}\right)}{\partial \Psi_{kn}} = 0 \Longrightarrow \Psi_{kn} = \frac{\Psi_{(k-1)n} + \Psi_{(k+1)n}}{2}, k = 1, \ldots, K_1 - 1,$$

so that the first partial derivatives are zero when the break points in the first slippage region are equidistant from each other, implying that:

$$\widetilde{\Psi}_{kn} = \frac{k}{K_1} X_n^{\text{cut}}, k = 0, \ldots, K_1. \qquad \text{[C.20]}$$

Average absolute error, equation [C.18], at the point $\widetilde{\Psi}_{\bullet n}$ is then

$$\begin{aligned}
\mathfrak{A}\left(\widetilde{\Psi}_{\bullet n}\right) &= \frac{a_{1n}}{6X_n^{\text{cut}}} \sum_{k=1}^{K_1} \left[\left(\frac{k}{K_1} - \frac{k-1}{K_1} \right) X_n^{\text{cut}} \right]^3 \\
&= \frac{a_{1n}}{6X_n^{\text{cut}}} \left(\frac{X_n^{\text{cut}}}{K_1} \right)^3 \sum_{k=1}^{K_1} 1 \\
\mathfrak{A}\left(\widetilde{\Psi}_{\bullet n}\right) &= \frac{a_{1n}}{6} \left(\frac{X_n^{\text{cut}}}{K_1} \right)^2. \qquad \text{[C.21]}
\end{aligned}$$

The Hessian matrix for $\mathfrak{A}\left(\Psi_{\bullet n}\right)$ is examined to determine if the point $\left[\widetilde{\Psi}_{\bullet n}, \mathfrak{A}\left(\widetilde{\Psi}_{\bullet n}\right)\right]$ defined by equations [C.20] and [C.21] is a maximum, minimum, or saddle point. Second

partial derivatives of $\mathfrak{A}(\Psi_{\bullet n})$ for $k = 1, \ldots, K_1 - 1$ are derived from equation [C.19] as:

$$\frac{\partial^2 \mathfrak{A}(\Psi_{\bullet n})}{\partial \Psi_{jn} \partial \Psi_{kn}} = \begin{cases} -\frac{a_{1n}}{X_n^{\text{cut}}} \left(\Psi_{kn} - \Psi_{(k-1)n} \right), & j = k-1, \\ \frac{a_{1n}}{X_n^{\text{cut}}} \left(\Psi_{(k+1)n} - \Psi_{(k-1)n} \right), & j = k, \\ -\frac{a_{1n}}{X_n^{\text{cut}}} \left(\Psi_{(k+1)n} - \Psi_{kn} \right), & j = k+1, \\ 0, & \text{otherwise.} \end{cases}$$

The resulting Hessian matrix, $\mathbf{H}\left[\mathfrak{A}(\Psi_{\bullet n})\right] \in \mathbb{R}^{(K_1-1)\times(K_1-1)}$, for general $\Psi_{\bullet n}$ is shown in Figure C.1. When evaluated at the point $\Psi_{\bullet n} = \widetilde{\Psi}_{\bullet n}$ as defined by equation [C.20], the Hessian matrix is:

$$\mathbf{H}\left[\mathfrak{A}\left(\widetilde{\Psi}_{\bullet n}\right)\right] = \frac{a_{1n}}{X_n^{\text{cut}}} \begin{bmatrix} 2 & -1 & & & & & & & \\ -1 & 2 & -1 & & & & & & \\ & & \ddots & & & & & & \\ & & & -1 & 2 & -1 & & & \\ & & & & & \ddots & & & \\ & & & & & & -1 & 2 & -1 \\ & & & & & & & -1 & 2 \end{bmatrix}.$$

Define $\widetilde{\mathbf{H}}$ as $\widetilde{\mathbf{H}} = \mathbf{H}\left[\mathfrak{A}\left(\widetilde{\Psi}_{\bullet n}\right)\right]$ and let $\mathbf{u} \in \mathbb{R}^{(K_1-1)}$. Then $\widetilde{\mathbf{H}}\mathbf{u}$ is:

$$\widetilde{\mathbf{H}}\mathbf{u} = \frac{a_{1n}}{X_n^{\text{cut}}} \begin{bmatrix} 2u_1 - u_2 \\ -u_1 + 2u_2 - u_3 \\ \vdots \\ -u_{k-1} + 2u_k - u_{k+1} \\ \vdots \\ -u_{K_1-3} + 2u_{K_1-2} - u_{K_1-1} \\ -u_{K_1-2} + 2u_{K_1-1} \end{bmatrix},$$

$$H\left[\mathfrak{A}\left(\Psi_{\bullet n}\right)\right]=\frac{a_{1n}}{X_n^{\text{cut}}}\begin{bmatrix}
\Psi_{2n} & \Psi_{1n}-\Psi_{2n} & & & & & & & \\
\Psi_{1n}-\Psi_{2n} & \Psi_{3n}-\Psi_{1n} & \Psi_{2n}-\Psi_{3n} & & & & & & \\
 & \bullet & & & & & & & \\
 & & \bullet & & & & & & \\
 & & & \bullet & & & & & \\
 & & & \Psi_{(k-1)n}-\Psi_{kn} & \Psi_{(k+1)n}-\Psi_{(k-1)n} & \Psi_{kn}-\Psi_{(k+1)n} & & & \\
 & & & & \bullet & & & & \\
 & & & & & \bullet & & & \\
 & & & & & & \bullet & & \\
 & & & & & & \Psi_{(K_1-3)n}-\Psi_{(K_1-2)n} & \Psi_{(K_1-1)n}-\Psi_{(K_1-3)n} & \Psi_{(K_1-2)n}-\Psi_{(K_1-1)n} \\
 & & & & & & & \Psi_{(K_1-2)n}-\Psi_{(K_1-1)n} & X_n^{\text{cut}}-\Psi_{(K_1-2)n}
\end{bmatrix}$$

Figure C.1: Hessian Matrix for the Average Absolute Error Function

and $\mathbf{u}'\widetilde{\mathbf{H}}\mathbf{u}$ is therefore:

$$\begin{aligned}
\frac{X_n^{\text{cut}}}{a_{1n}}\mathbf{u}'\widetilde{\mathbf{H}}\mathbf{u} &= (2u_1 - u_2)\,u_1 + (-u_1 + 2u_2 - u_3)\,u_2 + \cdots + (-u_{k-1} + 2u_k - u_{k+1})\,u_k \\
&\quad + \cdots + (-u_{K_1-3} + 2u_{K_1-2} - u_{K_1-1})\,u_{K_1-2} + (-u_{K_1-2} + 2u_{K_1-1})\,u_{K_2-1} \\
&= 2u_1^2 - 2u_1u_2 + 2u_2^2 - 2u_2u_3 + \cdots + 2u_k^2 - 2u_ku_{k+1} \\
&\quad + \cdots + 2u_{K_1-2}^2 - 2u_{K_1-2}u_{K_1-1} + 2u_{K_1-1}^2 \\
&= u_1^2 + \left(u_1^2 - 2u_1u_2 + u_2^2\right) + \cdots + \left(u_k^2 - 2u_ku_{k+1} + u_{k+1}^2\right) \\
&\quad + \cdots + \left(u_{K_1-2}^2 - 2u_{K_1-2}u_{K_1-1} + u_{K_1-1}^2\right) + u_{K_1-1}^2 \\
\frac{X_n^{\text{cut}}}{a_{1n}}\mathbf{u}'\widetilde{\mathbf{H}}\mathbf{u} &= u_1^2 + u_{K_1-1}^2 + \sum_{k=1}^{K_1-2}\left(u_k - u_{k+1}\right)^2.
\end{aligned}$$

Then, since $a_{1n} > 0$ and $X_n^{\text{cut}} > 0$ by the general conditions [C.2]:

$$\mathbf{u}'\widetilde{\mathbf{H}}\mathbf{u} = \frac{a_{1n}}{X_n^{\text{cut}}}\left(u_1^2 + u_{K_1-1}^2 + \sum_{k=1}^{K_1-2}\left(u_k - u_{k+1}\right)^2\right) > 0\;\forall\,\mathbf{u} \neq \mathbf{0},$$

and the Hessian matrix, $\widetilde{\mathbf{H}} = \mathbf{H}\left[\mathfrak{A}\left(\widetilde{\mathbf{\Psi}}_{\bullet n}\right)\right]$, is positive definite by definition. Hence, $\widetilde{\mathbf{\Psi}}_{\bullet n}$ is a strict local minimum over the first slippage region (e.g., see Theorem 4.1.4 in Bazaraa, Sherali, and Shetty [7, page 134]). Note, also, that no point, $\mathbf{\Psi}_{\bullet n}$, on a border (i.e., where one or more break points coincide) can yield an average absolute error less than or equal to $\mathfrak{A}\left(\widetilde{\mathbf{\Psi}}_{\bullet n}\right)$. Border points generate less than K_1 PWL segments and the resulting average absolute error will necessarily be larger than that for K_1 segments. Furthermore, the point $\widetilde{\mathbf{\Psi}}_{\bullet n}$ defined by equation [C.20] is the only local minimum for $\mathfrak{A}\left(\Psi_{\bullet n}\right)$ since it is the unique solution to equation [C.19] (e.g., see Theorem 4.1.3 in Bazaraa et al. [7, page133]).

Therefore, $\widetilde{\Psi}_{\bullet n}$ is a strict global minimum for the average absolute error function $\mathfrak{A}(\Psi_{\bullet n})$, equation [C.18].

C.4 Application Under General Conditions

Application of the maximum relative error and minimum average absolute error procedures is straightforward if the general conditions [C.2] hold. The maximum relative error procedure is always applied to the second slippage region. That procedure is also used in the first slippage region if a positive desired maximum relative error, ς_1, is specified. The minimum average absolute error procedure is applied in the first region when:

1. $\varsigma_1 \leq 0$, or
2. $\varsigma_1 > 0$ and the maximum relative error procedure does not yield a break point at the region boundary, X_n^{cut}.

A problematic situation occurs, however, if the slope adjustment factor is negative, i.e., $a_{2n} < 0$. The slippage function, $\widetilde{\mathfrak{S}}_n(x_n)$, is not convex across the region boundary when $a_{2n} < 0$. Break points may be assigned such that the last PWL slope in the first region is greater than one or more slopes in the second region when the slippage function is not convex across the region boundary. For instance, assuming that $\Phi_{(K_1-1)n} \leq \Phi_{K_1 n}$,

using equations [C.6a-b] and [C.3], and recalling that $\Psi_{K_1 n} = X_n^{\text{cut}}$ implies:

$$\begin{aligned}
&\Phi_{(K_1-1)n} \leq \Phi_{K_1 n} \\
&\Longrightarrow (a_{1n} + a_{2n}) \left(X_n^{\text{cut}} + \Psi_{(K_1+1)n}\right) + \kappa_n - a_{2n} X_n^{\text{cut}} \leq a_{1n} \left(\Psi_{(K_1-1)n} + X_n^{\text{cut}}\right) + \kappa_n \\
&\Longrightarrow (a_{1n} + a_{2n}) \Psi_{(K_1+1)n} \leq a_{1n} \Psi_{(K_1-1)n} \\
&\Longrightarrow \Psi_{(K_1+1)n} \leq \frac{a_{1n}}{a_{1n} + a_{2n}} \Psi_{(K_1-1)n} \qquad \text{[C.22]}
\end{aligned}$$

The relation depicted in the last expression above is clearly possible when $a_{2n} < 0$ (a_{2n} is never allowed to be less than or equal to $-a_{1n}$). This situation is illustrated by the graph in Figure C.2,

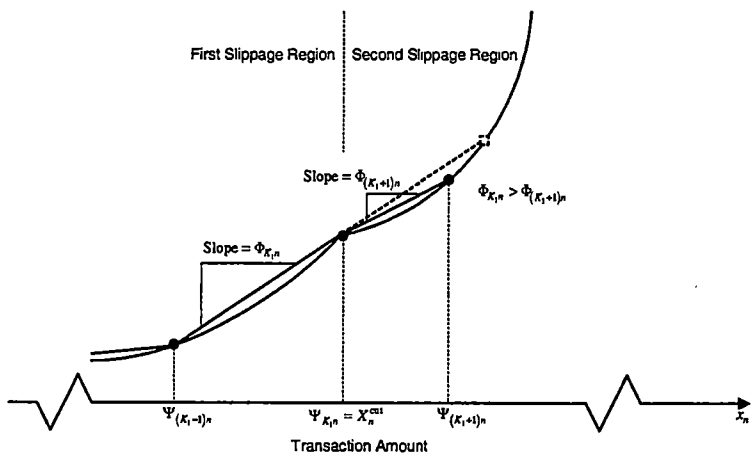


Figure C.2: PWL Slopes at the Region Boundary When $a_{2n} < 0$

and it can occur regardless of the procedure used to assign break points in the first slippage region. Non-increasing slopes can cause erroneous results in the market investment model and must, therefore, be prevented. If relation [C.22] is found to be true after $\Psi_{(K_1+1)n}$ is

determined, then that break point is adjusted according to:

$$\Psi_{(K_1+1)n} = (1+\delta)\frac{a_{1n}}{a_{1n}+a_{2n}}\Psi_{(K_1-1)n}$$

for some small positive value of δ. Note, however, that this may cause the maximum relative error observed in the first PWL segment of the second slippage region to exceed the specified desired value, ς_2.

Piece-wise linear approximation slopes, Φ_{kn}, $k = 1, \ldots, K$, are determined using equations [C.6a] and [C.3] once each applicable pair of break points have been set. The final break point is changed to positive infinity, $\Psi_{Kn} = \infty$, after the final slope, Φ_{Kn}, is evaluated.

C.5 Application Under Special Conditions

Several special conditions are allowed for by the coded procedures. For instance:

1. a no slippage model (no transaction costs) is specified by setting the number of break points in each slippage region to zero, $K_1 = K_2 = 0$,
2. a single slippage region model is designated when:
   (a) there is no region boundary, i.e., $X_n^{\text{cut}} = 0$, in which case the maximum relative error procedure is applied for all transaction amounts using the second set of values in equation [C.3], or

(b) the number of break points in the second slippage region is set to zero, $K_2 = 0$, in which case the method in 2(a) is applied except with the first set of values in equation [C.3], or

(c) the slope adjustment factor is set to zero, $a_{2n} = 0$, in which case the slippage function is the same for both slippage regions.

One special condition that requires individual attention is the case of no minimum slippage per position cost (i.e., $\kappa_n = 0$) when the maximum relative error procedure is specified for the first slippage region (i.e., $\varsigma_1 > 0$). Equation [C.16] indicates that the maximum relative error in the first PWL segment approaches infinity as κ_n approaches zero:

$$\lim_{\kappa_n \downarrow 0} \widehat{\mathcal{R}}_n(0) = \lim_{\kappa_n \downarrow 0} \frac{a_{1n}\Psi_{1n}}{\kappa_n} = \infty.$$

Therefore, equation [C.17] cannot be used in this case to determine the first break point, Ψ_{1n}. This problem is circumvented by assigning the first break point based upon the maximum absolute error versus the maximum relative error. Maximum absolute error will occur at the midpoint of the first PWL segment when $\kappa_n = 0$ since equations [C.3] through [C.7] indicate that:

$$\begin{aligned}
\frac{\partial \mathfrak{R}_n(x_n)}{\partial x_n} &= \frac{\partial}{\partial x_n}\left[\widehat{\mathfrak{S}}_{kn}(x_n) - \widetilde{\mathfrak{S}}_n(x_n)\right] \\
&= \frac{\partial}{\partial x_n}\left[\Phi_{1n}x_n + v_{1n} - x_n\left(\tilde{a}_{1n}x_n + \tilde{a}_{2n}\right)\right] \\
&= \frac{\partial}{\partial x_n}\left[a_{1n}\Psi_{1n}x_n - a_{1n}x_n^2\right] \\
\frac{\partial \mathfrak{R}_n(x_n)}{\partial x_n} &= a_{1n}\Psi_{1n} - 2a_{1n}x_n.
\end{aligned}$$

Setting the above derivative to zero yields:

$$x_n = \frac{1}{2}\Psi_{1n}$$

which must be a maximum since the second derivative $(-2a_{1n})$ is negative indicating that $\Re_n(x_n)$ is a concave function over the domain of interest. The maximum absolute error at the midpoint of the first PWL segment is:

$$\Re_n\left(\frac{1}{2}\Psi_{1n}\right) = a_{1n}\Psi_{1n}\left(\frac{1}{2}\Psi_{1n}\right) - a_{1n}\left(\frac{1}{2}\Psi_{1n}\right)^2 = \frac{a_{1n}}{4}\Psi_{1n}^2.$$

The first break point is then set so that the maximum absolute error in the first segment is equal to the specified desired maximum relative error:

$$\widetilde{\Psi}_{1n} = 2\sqrt{\frac{\varsigma_1}{a_{1n}}}.$$

C.6 PWL Parameters for Quadratic Downside Deviation

Piece-wise linear approximation slopes and break points for quadratic downside deviation are also determined with the maximum relative error and minimum average absolute error procedures. Using these procedures is justified since quadratic downside deviation can be considered a special case of the quadratic slippage function by setting $a_{1n} = 1$ and $a_{2n} = \kappa_n = 0$. The minimum average absolute error procedure can be applied by creating two artificial downside deviation regions with a user specified value for the simulated region boundary.

Appendix D
Sizes of Model MIMPSLP Problems

Equations for the sizes for model **MIMPSLP** problems are derived in this appendix based upon four parameters: number of variables, constraints, non-zero technology matrix coefficients, and nodes in the decision tree. Equations for the number of nodes in a given period and the number of cumulative nodes though a given period are listed in the first section below. The number of primal constraints and dual variables are derived in the second section. The third section contains derivations for the number of primal variables and dual constraints. Sizes for the technology matrices (number of non-zeros) are detailed in the fourth section. Comments concerning problem size appear in the fifth and final section of this appendix. Size results are summarized in Table 5.3 on page 204 for easy reference.

D.1 Number of Nodes in the Decision Tree

Based upon definitions in Section 2.2.1, the number of nodes, H_t, in period t and the number of cumulative nodes, $H^{(t)}$, through period t, $1 \leq t \leq T$, are:

$$H_t = \prod_{j=1}^{t-1} L_j, \text{ where } H_1 = \prod_{j=1}^{0} L_j = 1 \qquad \text{[D.1]}$$

and

$$H^{(t)} = \sum_{j=1}^{t} H_j = \sum_{j_1=1}^{t} \left(\prod_{j_2=1}^{j_1-1} L_{j_2} \right). \qquad \text{[D.2]}$$

The remaining three size parameters can then be easily determined by summing the results for a generic node over all nodes in the decision tree.

Let $\#_V^P$ and $\#_Z^P$ be the number of variables and technology matrix non-zeros respectively for problem **PMPGLP** [5.19] on page 184. Let $\#_C^P$ ($\#_{C^*}^P$) be the number of constraints for problem **PMPGLP** when upper bounds on the decision variables are treated as bounds (constraints). Denote the corresponding size parameters for some node $[\bullet]_t$ in the decision tree by $\#_V^{P[\bullet]_t}$, $\#_C^{P[\bullet]_t}$ ($\#_{C^*}^{P[\bullet]_t}$), and $\#_Z^{P[\bullet]_t}$. Size parameters for the dual problem **DMPGLP** [5.24] on page 188 are represented similarly except that superscript P is replaced with a D. Equations for each of the remaining three size parameters are derived below followed by comments regarding the sizes of model problems.

D.2 Primal Constraints – Dual Variables

One budget constraint, $2N$ slippage constraints (N buy and N sell constraints), and L_t deviation constraints are associated with each node in the primal problem **PMPGLP** [5.19] when upper bounds on decision variables are treated as bounds and not constraints. Therefore, for each node $[\bullet]_t = (t, h_t)$, $h_t = 1, \ldots, H_t$, $t = 1, \ldots, T$,

$$\#_C^{P[\bullet]_t} = 1 + 2N + L_t$$

which, with equations [D.1] and [D.2], implies that:

$$\begin{aligned}
\#_{\mathrm{C}}^{\mathrm{P}} &= \sum_{t=1}^{T} H_t \#_{\mathrm{C}}^{\mathrm{P}[\bullet]_t} \\
&= \sum_{t=1}^{T} H_t \left(1 + 2N + L_t\right) \\
&= \left(1 + 2N\right) \sum_{t=1}^{T} H_t + \sum_{t=1}^{T} H_t L_t \\
\#_{\mathrm{C}}^{\mathrm{P}} &= \left(1 + 2N\right) H^{(T)} + \sum_{t=1}^{T} H_t L_t. \qquad \text{[D.3]}
\end{aligned}$$

The number of constraints is significantly larger when upper bounds on decision variables are treated explicitly as constraints. There are $(K-1)\,N$ upper bounding constraints on primal slippage variables and $(Q-1)\,L_t$ upper bounding constraints on primal deviation variables. In addition, there is an upper bounding constraint on each translated portfolio variable that has a finite upper bound. No upper bounding constraints for portfolio variables are accounted for since all problem instances analyzed in Chapter 6 assume unbounded portfolio variables. The number of constraints at some node in this case is then:

$$\#_{\mathrm{C}^*}^{\mathrm{P}[\bullet]_t} = 1 + 2N + (K-1)N + L_t + (Q-1)L_t = 1 + (K+1)N + QL_t,$$

which implies that:

$$\begin{aligned}
\#^{\mathrm{P}}_{\mathrm{C}^*} &= \sum_{t=1}^{T} H_t \#^{\mathrm{P}[\bullet]_t}_{\mathrm{C}^*} \\
&= \sum_{t=1}^{T} H_t \left[1 + (K+1)N + QL_t\right] \\
&= \left[1 + (K+1)N\right] \sum_{t=1}^{T} H_t + Q \sum_{t=1}^{T} H_t L_t \\
\#^{\mathrm{P}}_{\mathrm{C}^*} &= \left[1 + (K+1)N\right] H^{(T)} + Q \sum_{t=1}^{T} H_t L_t. \qquad \text{[D.4]}
\end{aligned}$$

Dual problem **DMPGLP** [5.24] is based upon employing upper bounding constraints in the primal problem so that the number of dual variables is:

$$\#^{\mathrm{D}}_{\mathrm{V}} = \#^{\mathrm{P}}_{\mathrm{C}^*} = \left[1 + (K+1)N\right] H^{(T)} + Q \sum_{t=1}^{T} H_t L_t. \qquad \text{[D.5]}$$

D.3 Primal Variables – Dual Constraints

There are N translated portfolio variables, KN primal slippage variables, and QL_t primal deviation variables associated with each node for problem **PMPGLP** [5.19]. Therefore, for each node $[\bullet]_t = (t, h_t)$, $h_t = 1, \ldots, H_t$, $t = 1, \ldots, T$,

$$\#^{\mathrm{P}[\bullet]_t}_{\mathrm{V}} = N + KN + QL_t = (K+1)\,N + QL_t$$

which, with equations [D.1] and [D.2], implies that

$$\begin{aligned}
\#_{\mathrm{V}}^{\mathrm{P}} &= \sum_{t=1}^{T} H_t \#_{\mathrm{V}}^{\mathrm{P}[\bullet]_t} \\
&= \sum_{t=1}^{T} H_t \left[(K+1) N + Q L_t \right] \\
&= (K+1) N \sum_{t=1}^{T} H_t + Q \sum_{t=1}^{T} H_t L_t \\
\#_{\mathrm{V}}^{\mathrm{P}} &= (K+1) N H^{(T)} + Q \sum_{t=1}^{T} H_t L_t. \qquad \text{[D.6]}
\end{aligned}$$

The number of constraints in dual problem **DMPGLP** [5.24] is then

$$\#_{\mathrm{C}}^{\mathrm{D}} = \#_{\mathrm{V}}^{\mathrm{P}} = (K+1) N H^{(T)} + Q \sum_{t=1}^{T} H_t L_t. \qquad \text{[D.7]}$$

D.4 Non-Zero Technology Matrix Coefficients

The number of non-zero technology matrix coefficients are first determined for problem **PMPGLP** [5.19] assuming no upper bounding constraints on the decision variables. The number of non-zero coefficients at each node by constraint type is:

budget: N,

slippage: $(1+K)$ if $t = 1$ or $(2+K)$ is $2 \leq t \leq T$, and

deviation: $(N+Q)$.

Therefore, for each node $[\bullet]_t = (t, h_t)$, $h_t = 1, \ldots, H_t$, $t = 1, \ldots, T$,

$$\#_{\mathrm{Z}}^{\mathrm{P}[\bullet]_t} = N + 2N(j+K) + L_t(N+Q),$$

where $j = 1$ if $t = 1$ or $j = 2$ otherwise. Equations [D.1] and [D.2] then imply that

$$\begin{aligned}
\#_Z^P &= \sum_{t=1}^{T} H_t \#_Z^{P[\bullet]_t} \\
&= \#_Z^{P[]} + \sum_{t=2}^{T} H_t \#_Z^{P[\bullet]_t} \\
&= N + 2N(1+K) + L_1(N+Q) + \sum_{t=2}^{T} H_t [N + 2N(2+K) + L_t(N+Q)] \\
&= [1 + 2(1+K)] N + [N + 2N(2+K)] \sum_{t=2}^{T} H_t + (N+Q) \sum_{t=1}^{T} H_t L_t \\
&= [1 + 2(1+K)] N + [1 + 2(2+K)] N \left(H^{(T)} - 1\right) + (N+Q) \sum_{t=1}^{T} H_t L_t \\
\#_Z^P &= \left[(2K+5) H^{(T)} - 2\right] N + (N+Q) \sum_{t=1}^{T} H_t L_t. \qquad \text{[D.8]}
\end{aligned}$$

There are an additional $(K-1)N$ non-zero coefficients for slippage and $(Q-1)L_t$ non-zero coefficients for deviation at each node when these two sets of decision variables have explicit upper bounding constraints. Therefore, the number of non-zeros in the technology coefficient matrix for dual problem **DMPGLP** is:

$$\begin{aligned}
\#_{\mathrm{Z}}^{\mathrm{D}} &= \#_{\mathrm{Z}}^{\mathrm{P}} + \sum_{t=1}^{T} H_t \left[(K-1)\, N + (Q-1)\, L_t \right] \\
&= \#_{\mathrm{Z}}^{\mathrm{P}} + (K-1)\, N \sum_{t=1}^{T} H_t + (Q-1) \sum_{t=1}^{T} H_t L_t \\
&= \#_{\mathrm{Z}}^{\mathrm{P}} + (K-1)\, N H^{(T)} + (Q-1) \sum_{t=1}^{T} H_t L_t \\
&= \left[(2K+5)\, H^{(T)} - 2 + (K-1)\, H^{(T)} \right] N + (N+Q+Q-1) \sum_{t=1}^{T} H_t L_t \\
\#_{\mathrm{Z}}^{\mathrm{D}} &= \left[(3K+4)\, H^{(T)} - 2 \right] N + (N+2Q-1) \sum_{t=1}^{T} H_t L_t. \qquad \text{[D.9]}
\end{aligned}$$

D.5 Comments

Model **MIMPSLP** problems, and stochastic linear programs in general, can obviously be extremely large. For instance, the cumulative number of nodes in the model **MIMPSLP** decision tree with the same number of random outcomes, say L, used in each period is (using equation [D.2]):

$$H^{(T)} = \sum_{t=1}^{T} H_t = \sum_{t=1}^{T} \left(\prod_{j=1}^{t-1} L \right) = \sum_{t=1}^{T} L^{t-1},$$

which grows exponentially with L when multiple periods are involved and $L > 1$. Even a problem involving only a few periods, say 2-4, could be so large that it is impossible to store problem data in the active memory of a computer. Decomposition solution procedures help to alleviate complications caused by problem size since they rarely require access to

all problem data at any one time. Greater detail on this topic is given in Chapter 6 with the application of decomposition procedures to model **MIMPSLP** problems.

Appendix E
Decomposition of Model MIMPSLP Problems

This appendix contains special problem formulations and procedures used to solve model **MIMPSLP** problems. Problem formulations for L-Shaped, Dantzig-Wolfe, and myopic decomposition are derived in the first section below. Three types of subproblems – slippage, deviation, and nodal – incorporated by LSD and DWD are also introduced and defined in the first section. Procedures used to solve the slippage and deviation subproblems are detailed in the second and third sections respectively. The fourth and final section contains detailed procedures used to determine upper bounds for the nodal subproblems.

Note that period-index and path vector node labels as defined in Section 2.2.1 are used interchangeably below. Equation [2.7] on page 28 can be used to determine the path vector label, $[\bullet]_t = [l_1, \ldots, l_{t-1}]$, for a node given the period index label, (t, h_t), for that node.

E.1 Decomposition Procedures

Each of the solution techniques summarized in Section 5.4 are described in detail below. The first five subsections are devoted to the application of Dantzig-Wolfe and/or L-Shaped decomposition to model problems. Master problems and subproblem types are described in the first two subsections. These descriptions are for example problems based upon as-

sumed values for the model master strategy, minor strategy, and tactics parameters as described in Section 4.4.3. Problem structuring and DWD/LSD application techniques for single-period, two-stage, and multi-stage problems are covered in the next three subsections. Myopic decomposition of model **MIMPSLP** problems is discussed in the sixth and final subsection.

E.1.1 DWD/LSD Master Problems

The problem descriptions given below apply to the single first stage master problem of a Dantzig-Wolfe or L-Shaped decomposition procedure. Formulations for the RMP-SUB problems in the nested decomposition of a multi-stage problem are similar and are described in Section E.1.5. Three different types of decomposition subproblems are also introduced below. Subproblem types are described in detail in Section E.1.2. All DWD/LSD procedures are based upon using the block-separable recourse property of model **MIMPSLP** problems.

L-Shaped decomposition relaxed master problems are initiated with the primal budget constraints since these constraints involve only the aggregate level translated portfolio variables, $x^{(0)[\bullet]_t}$. Similarly, Dantzig-Wolfe decomposition restricted master problems are initiated with the dual budget variables (i.e., the dual multipliers to the primal budget constraints). The first stage relaxed master problem, **LSD RMP**, for L-Shaped decomposition is derived below. The corresponding restricted master problem, **DWD RMP**, for Dantzig-Wolfe decomposition is then the dual to **LSD RMP**.

Appendix E Decomposition of Model MIMPSLP Problems

Recall from Section 4.4.3 that the structure for the RMP is determined by the value of the major strategy. Using the notation from that section, assume that the major strategy sets $\bar{t}$, $1 \leq \bar{t} \leq T$, referred to as the *cutoff* period, as the index for the last period that is to contribute budget constraints to the initial RMP. All primal budget constraints in periods 1 through $\bar{t}$ are then accounted for in the RMP while the primal slippage and deviation constraints at the nodes in these periods must be accounted for in one or more subproblems. In addition, if $\bar{t} < T$, then all constraints at each node in periods $\hat{t} = \bar{t} + 1$ through T must also be accounted for in subproblems. The number of subproblems to create and the manner in which subproblem solution information is used to generate additional cuts for the RMP are determined respectively by the values of the minor strategy and tactics as described in Section 4.4.3. Assume that the minor strategy dictates that the slippage and deviation constraints in periods 1 through $\bar{t}$ be used to create $J_{\mathrm{S}}^{[\bullet]_t}$ *slippage* and $J_{\mathrm{D}}^{[\bullet]_t}$ *deviation* subproblems at each node $[\bullet]_t$ for $1 \leq t \leq \bar{t}$. Assume that the minor strategy also requires that all constraints at nodes in periods $\hat{t}$ through T be used to create $J_{\mathrm{N}}^{(\hat{t})}$ *nodal* subproblems when $\bar{t} < T$. Furthermore, assume that the value of tactics dictates that an optimality cut is generated for each subproblem. This information is used below to formulate the initial RMP while details on the formulations of subproblems and generation of cuts are given in Section E.1.2.

Let G index the current iteration of the DWD/LSD algorithm and let $K_{\mathrm{S}}^{[\bullet]_t(G)}$, $K_{\mathrm{D}}^{[\bullet]_t(G)}$, and $K_{\mathrm{N}}^{(\hat{t})(G)}$ be the number of cuts at the beginning of iteration G that correspond to slippage, deviation, and nodal subproblems respectively. Denote the slippage, deviation, and

nodal relaxation variables by $\boldsymbol{\theta}_{\mathrm{S}}^{[\bullet]_t} \in \mathbb{R}^{J_{\mathrm{S}}^{[\bullet]_t}}$, $\boldsymbol{\theta}_{\mathrm{D}}^{[\bullet]_t} \in \mathbb{R}^{J_{\mathrm{D}}^{[\bullet]_t}}$, and $\boldsymbol{\theta}_{\mathrm{N}}^{(\hat{t})} \in \mathbb{R}^{J_{\mathrm{N}}^{(\hat{t})}}$ respectively. The initial RMP in array notation at iteration $G = 1$ of the LSD algorithm is then **LSD RMP**(1):

$$z_{\mathrm{UB}}^{(1)} = \max \sum_{t=1}^{\bar{t}} \sum_{h_t=1}^{H_t} \left(\mathbf{c}'^{(0)[\bullet]_t} \mathbf{x}^{(0)[\bullet]_t} + \mathbf{1}'\boldsymbol{\theta}_{\mathrm{S}}^{[\bullet]_t} + \mathbf{1}'\boldsymbol{\theta}_{\mathrm{D}}^{[\bullet]_t} \right) + \mathbf{1}'\boldsymbol{\theta}_{\mathrm{N}}^{(\hat{t})} \qquad \text{[E.1]}$$

$$\begin{aligned}
\text{s.t. } \mathbf{W}^{(0,0)[\bullet]_t} \mathbf{x}^{(0)[\bullet]_t} &\leq \mathbf{b}^{(0)[\bullet]_t}, \quad h_t = 1, \ldots, H_t, \quad t = 1, \ldots, \bar{t}, \\
\boldsymbol{\theta}_{\mathrm{S}}^{[\bullet]_t} &\leq \mathbf{0}, \quad h_t = 1, \ldots, H_t, \quad t = 1, \ldots, \bar{t}, \\
\boldsymbol{\theta}_{\mathrm{D}}^{[\bullet]_t} &\leq \mathbf{0}, \quad h_t = 1, \ldots, H_t, \quad t = 1, \ldots, \bar{t}, \\
\boldsymbol{\theta}_{\mathrm{N}}^{(\hat{t})} &\leq \boldsymbol{\theta}_{\mathrm{N}}^{(\hat{t})\max}, \\
\mathbf{x}^{(0)[\bullet]_t} &\geq \mathbf{0}, \quad h_t = 1, \ldots, H_t, \quad t = 1, \ldots, \bar{t}, \\
\boldsymbol{\theta}_{\mathrm{S}}^{[\bullet]_t} &\quad \text{free}, \quad h_t = 1, \ldots, H_t, \quad t = 1, \ldots, \bar{t}, \\
\boldsymbol{\theta}_{\mathrm{D}}^{[\bullet]_t} &\quad \text{free}, \quad h_t = 1, \ldots, H_t, \quad t = 1, \ldots, \bar{t}, \\
\boldsymbol{\theta}_{\mathrm{N}}^{(\hat{t})} &\quad \text{free},
\end{aligned}$$

based upon problem **PMPGLP** formulation [5.46] on page 201. Note that initial upper bounding cuts for each relaxation variable are present in the formulation so that $K_{\mathrm{S}}^{[\bullet]_t(1)} = 1$, $K_{\mathrm{D}}^{[\bullet]_t(1)} = 1$, and $K_{\mathrm{N}}^{(\hat{t})(1)} = 1$ for all subproblems. These upper bounding cuts are validated, and $\boldsymbol{\theta}_{\mathrm{N}}^{(\hat{t})\max}$ defined, in the following subsection. The scalar formulation for **LSD RMP**(1) is:

$$z_{\mathrm{UB}}^{(1)} = \max \sum_{t=1}^{\bar{t}} \sum_{h_t=1}^{H_t} \left(\sum_{n=1}^{N} \mu_n^{[\bullet]_t} x_n^{(0)[\bullet]_t} + \sum_{j=1}^{J_{\mathrm{S}}^{[\bullet]_t}} \theta_{\mathrm{S}j}^{[\bullet]_t} + \sum_{j=1}^{J_{\mathrm{D}}^{[\bullet]_t}} \theta_{\mathrm{D}j}^{[\bullet]_t} \right) + \sum_{j=1}^{J_{\mathrm{N}}^{(\hat{t})}} \theta_{\mathrm{N}j}^{(\hat{t})} \qquad \text{[E.2]}$$

$$
\begin{aligned}
\text{s.t.} \quad \sum_{n=1}^{N} \sigma_n^{[\bullet]_t} x_n^{(0)[\bullet]_t} &\leq \hat{\beta}^{[\bullet]_t}, && h_t = 1,\ldots,H_t, \quad t = 1,\ldots,\bar{t}, \\
\theta_{\mathrm{S}j}^{[\bullet]_t} &\leq 0, && j = 1,\ldots,J_{\mathrm{S}}^{[\bullet]_t}, \quad h_t = 1,\ldots,H_t, \quad t = 1,\ldots,\bar{t}, \\
\theta_{\mathrm{D}j}^{[\bullet]_t} &\leq 0, && j = 1,\ldots,J_{\mathrm{D}}^{[\bullet]_t}, \quad h_t = 1,\ldots,H_t, \quad t = 1,\ldots,\bar{t}, \\
\theta_{\mathrm{N}j}^{(\hat{t})} &\leq \theta_{\mathrm{N}j}^{(\hat{t})\max}, && j = 1,\ldots,J_{\mathrm{N}}^{(\hat{t})}, \\
x_n^{(0)[\bullet]_t} &\geq 0, && n = 1,\ldots N, \quad h_t = 1,\ldots,H_t, \quad t = 1,\ldots,\bar{t}, \\
\theta_{\mathrm{S}j}^{[\bullet]_t} &\quad \text{free}, && j = 1,\ldots,J_{\mathrm{S}}^{[\bullet]_t}, \quad h_t = 1,\ldots,H_t, \quad t = 1,\ldots,\bar{t}, \\
\theta_{\mathrm{D}j}^{[\bullet]_t} &\quad \text{free}, && j = 1,\ldots,J_{\mathrm{D}}^{[\bullet]_t}, \quad h_t = 1,\ldots,H_t, \quad t = 1,\ldots,\bar{t}, \\
\theta_{\mathrm{N}j}^{(\hat{t})} &\quad \text{free}, && j = 1,\ldots,J_{\mathrm{N}}^{(\hat{t})},
\end{aligned}
$$

based upon problem **PMPGLP** formulation [5.19] on page 184.

The initial RMP for DWD using comparable decomposition strategy values is the dual of formulation [E.1] or [E.2]. Let

$$
\begin{aligned}
\boldsymbol{\eta}_{\mathrm{S}}^{[\bullet]_t(j)} &= \left(\eta_{\mathrm{S}1}^{[\bullet]_t(j)},\ldots,\eta_{\mathrm{S}K_{\mathrm{S}}^{[\bullet]_t(G)}}^{[\bullet]_t(j)}\right)', \; j = 1,\ldots,J_{\mathrm{S}}^{[\bullet]_t}, \\
\boldsymbol{\eta}_{\mathrm{D}}^{[\bullet]_t(j)} &= \left(\eta_{\mathrm{D}1}^{[\bullet]_t(j)},\ldots,\eta_{\mathrm{D}K_{\mathrm{D}}^{[\bullet]_t(G)}}^{[\bullet]_t(j)}\right)', \; j = 1,\ldots,J_{\mathrm{D}}^{[\bullet]_t}, \\
\boldsymbol{\eta}_{\mathrm{N}}^{(\hat{t})(j)} &= \left(\eta_{\mathrm{N}1}^{(\hat{t})(j)},\ldots,\eta_{\mathrm{N}K_{\mathrm{N}}^{(\hat{t})(G)}}^{(\hat{t})(j)}\right)', \; j = 1,\ldots,J_{\mathrm{N}}^{(\hat{t})},
\end{aligned}
\qquad \text{[E.3]}
$$

be the vectors of dual multipliers to the upper bounding constraints on the slippage, deviation, and nodal optimality cuts respectively at LSD iteration G. The initial RMP in array

notation at iteration $G = 1$ of the DWD algorithm is then **DWD RMP**(1):

$$z_{\mathrm{UB}}^{(1)} = \min \sum_{t=1}^{\bar{t}} \sum_{h_t=1}^{H_t} \mathrm{b}'^{(0)[\bullet]_t} \boldsymbol{\pi}^{(0)[\bullet]_t} + \sum_{j=1}^{J_{\mathrm{N}}^{(\hat{t})}} \theta_{\mathrm{N}j}^{(\hat{t})\max} \boldsymbol{\eta}_{\mathrm{N}}^{(\hat{t})(j)}$$

$$\begin{aligned}
\text{s.t.} \quad \mathbf{W}'^{(0,0)[\bullet]_t} \boldsymbol{\pi}^{(0)[\bullet]_t} &\geq \mathbf{c}^{(0)[\bullet]_t}, & h_t &= 1, \ldots, H_t, \\
& & t &= 1, \ldots, \bar{t}, \\
\boldsymbol{\eta}_{\mathrm{N}}^{(\hat{t})(j)} &= \mathbf{1}, & j &= 1, \ldots, J_{\mathrm{N}}^{(\hat{t})}, \\
\boldsymbol{\pi}^{(0)[\bullet]_t} &\geq \mathbf{0}, & h_t &= 1, \ldots, H_t, \\
& & t &= 1, \ldots, \bar{t}, \\
\boldsymbol{\eta}_{\mathrm{N}}^{(\hat{t})(j)} &\geq \mathbf{0}, & j &= 1, \ldots, J_{\mathrm{N}}^{(\hat{t})}.
\end{aligned} \qquad \text{[E.4]}$$

The scalar formulation for **DWD RMP**(1) is:

$$z_{\mathrm{UB}}^{(1)} = \min \sum_{t=1}^{\bar{t}} \sum_{h_t=1}^{H_t} \hat{\beta}^{[\bullet]_t} \pi_0^{(0)[\bullet]_t} + \sum_{j=1}^{J_{\mathrm{N}}^{(\hat{t})}} \theta_{\mathrm{N}j}^{(\hat{t})\max} \eta_{\mathrm{N1}}^{(\hat{t})(j)}$$

$$\begin{aligned}
\text{s.t.} \quad \sigma_n^{[\bullet]_t} \pi_0^{(0)[\bullet]_t} &\geq \hat{\varrho}_t \hat{p}^{[\bullet]_t} \mu_n^{[\bullet]_t}, & n &= 1, \ldots, N, \\
& & h_t &= 1, \ldots, H_t, \\
& & t &= 1, \ldots, \bar{t}, \\
\eta_{\mathrm{N1}}^{(\hat{t})(j)} &= 1, & j &= 1, \ldots, J_{\mathrm{N}}^{(\hat{t})}, \\
\pi_0^{(0)[\bullet]_t} &\geq 0.
\end{aligned} \qquad \text{[E.5]}$$

All model **MIMPSLP** problem instances defined in Chapter 6 have portfolio variables that are unbounded from above. Therefore, the corresponding dual multipliers, $\pi_n^{(0)[\bullet]_t}$, $n = 1, \ldots, N$, to the upper bounding constraints for portfolio variables at all nodes are zero and are omitted from the above formulations.

E.1.2 DWD/LSD Subproblems

One or more subproblems account for the constraints and objective function components that are not included in the initial RMP. The number of subproblems that are possible depends upon the degree of separability provided by the structure of the remainder of the problem. Derivations for the three types of subproblems, slippage, deviation, and nodal, introduced in the previous subsection are given below based upon that portion of problem **PMPGLP** that is not included in **LSD RMP**(1).

Let $\mathbf{X}_{\bar{t}}^{(0)} \in \mathbb{R}^{N \times H^{(\bar{t})}}$ be the matrix whose columns are the vectors of translated portfolio variables at all nodes in periods $t = 1, \ldots, \bar{t}$ so that:

$$\mathbf{X}_{\bar{t}}^{(0)} = \left[\mathbf{x}^{(0)\,(1,1)}, \mathbf{x}^{(0)\,(2,1)}, \ldots, \mathbf{x}^{(0)\,(2,H_2)}, \ldots, \mathbf{x}^{(0)\,(\bar{t},1)}, \ldots, \mathbf{x}^{(0)\,(\bar{t},H_{\bar{t}})}\right],$$

where period-index labels are used to identify nodes instead of path vector labels. The problem that results from the portion of problem **PMPGLP** that is not in the initial RMP is denoted by **LSD LEFT** and the associated second stage value function is denoted by $\Omega_2^{(\bar{t})}\left(\mathbf{X}_{\bar{t}}^{(0)}\right)$. Then, based upon the array notation formulations of problems **PMPGLP** [5.46] on page 201 and **LSD RMP**(1) [E.1] on page 365, **LSD LEFT** is:

$$\begin{aligned}\Omega_2^{(\bar{t})}\left(\mathbf{X}_{\bar{t}}^{(0)}\right) = \max & \sum_{t=1}^{\bar{t}} \sum_{h_t=1}^{H_t} \left(\mathbf{c}'^{(1)[\bullet]_t}\mathbf{x}^{(1)[\bullet]_t} + \mathbf{c}'^{(2)[\bullet]_t}\mathbf{x}^{(2)[\bullet]_t}\right) \\ & + \sum_{t=\hat{t}}^{T} \sum_{h_t=1}^{H_t} \mathbf{c}'^{(0)[\bullet]_t}\mathbf{x}^{(0)[\bullet]_t} + \sum_{t=\hat{t}}^{T} \sum_{h_t=1}^{H_t} \left(\mathbf{c}'^{(1)[\bullet]_t}\mathbf{x}^{(1)[\bullet]_t} + \mathbf{c}'^{(2)[\bullet]_t}\mathbf{x}^{(2)[\bullet]_t}\right)\end{aligned} \qquad \text{[E.6]}$$

s.t. detailed level constraints at nodes $[\bullet]_t = (t, h_t)$, $h_t = 1, \ldots, H_t$, $t = 1, \ldots, \bar{t}$,

$$\begin{aligned}
\mathbf{W}^{(1,1)[\bullet]_t}\mathbf{x}^{(1)[\bullet]_t} &\leq \mathbf{b}^{(1)[\bullet]_t} - \mathbf{W}^{(1,0)[\bullet]_t}\mathbf{x}^{(0)[\bullet]_t} - \mathbf{B}^{(1,0)[\bullet]_t}\mathbf{x}^{(0)[\bullet]_{t-1}}, \\
\mathbf{W}^{(2,2)[\bullet]_t}\mathbf{x}^{(2)[\bullet]_t} &\leq \mathbf{b}^{(2)[\bullet]_t} - \mathbf{W}^{(2,0)[\bullet]_t}\mathbf{x}^{(0)[\bullet]_t}, \\
\mathbf{x}^{(1)[\bullet]_t} &\geq \mathbf{0}, \\
\mathbf{x}^{(2)[\bullet]_t} &\geq \mathbf{0},
\end{aligned}$$

aggregate level constraints at nodes $[\bullet]_t = (t, h_t)\,,\ h_t = 1, \ldots, H_t,\ t = \hat{t}, \ldots, T,$

$$\begin{aligned}
\mathbf{W}^{(0,0)[\bullet]_t}\mathbf{x}^{(0)[\bullet]_t} &\leq \mathbf{b}^{(0)[\bullet]_t}, \\
\mathbf{x}^{(0)[\bullet]_t} &\geq \mathbf{0},
\end{aligned}$$

detailed level constraints at nodes $[\bullet]_{\hat{t}} = \left(\hat{t}, h_{\hat{t}}\right)\,,\ h_{\hat{t}} = 1, \ldots, H_{\hat{t}},$

$$\begin{aligned}
\mathbf{W}^{(1,0)[\bullet]_{\hat{t}}}\mathbf{x}^{(0)[\bullet]_{\hat{t}}} + \mathbf{W}^{(1,1)[\bullet]_{\hat{t}}}\mathbf{x}^{(1)[\bullet]_{\hat{t}}} &\leq \mathbf{b}^{(1)[\bullet]_{\hat{t}}} - \mathbf{B}^{(1,0)[\bullet]_{\hat{t}}}\mathbf{x}^{(0)[\bullet]_{\hat{t}}}, \\
\mathbf{W}^{(2,0)[\bullet]_{\hat{t}}}\mathbf{x}^{(0)[\bullet]_{\hat{t}}} \quad \mathbf{W}^{(2,2)[\bullet]_{\hat{t}}}\mathbf{x}^{(2)[\bullet]_{\hat{t}}} &\leq \mathbf{b}^{(2)[\bullet]_{\hat{t}}}, \\
\mathbf{x}^{(1)[\bullet]_{\hat{t}}} &\geq \mathbf{0}, \\
\mathbf{x}^{(2)[\bullet]_{\hat{t}}} &\geq \mathbf{0},
\end{aligned}$$

and detailed level constraints at nodes $[\bullet]_t = (t, h_t)\,,\ h_t = 1, \ldots, H_t,\ t = \hat{t}+1, \ldots, T,$

$$\begin{aligned}
\mathbf{B}^{(1,0)[\bullet]_t}\mathbf{x}^{(0)[\bullet]_{t-1}} + \mathbf{W}^{(1,0)[\bullet]_t}\mathbf{x}^{(0)[\bullet]_t} + \mathbf{W}^{(1,1)[\bullet]_t}\mathbf{x}^{(1)[\bullet]_t} &\leq \mathbf{b}^{(1)[\bullet]_t}, \\
\mathbf{W}^{(2,0)[\bullet]_t}\mathbf{x}^{(0)[\bullet]_t} + \mathbf{W}^{(2,2)[\bullet]_t}\mathbf{x}^{(2)[\bullet]_t} &\leq \mathbf{b}^{(2)[\bullet]_t}, \\
\mathbf{x}^{(1)[\bullet]_t} &\geq \mathbf{0}, \\
\mathbf{x}^{(2)[\bullet]_t} &\geq \mathbf{0}.
\end{aligned}$$

Problem **LSD LEFT** [E.6] can then be treated as a single subproblem or it can be separated into several subproblems due to the structures of the coefficient arrays in the constraints and in the objective function. Regardless of the number of subproblems designated by the decomposition strategy values, the most efficient method for solving **LSD LEFT** is to separate it into independent component problems. The resulting component problems

are referred to as *component subproblems* in order to distinguish them from *decomposition subproblems*. A decomposition subproblem may then coincide with a component subproblem or it may be a composite of two or more component subproblems. Problem **LSD LEFT**, for instance, is a composite of all component subproblems. Each component subproblem belongs to one of three types which correspond to the three types of decomposition subproblems introduced in the previous subsection: slippage, deviation, and nodal. Separation of problem **LSD LEFT** into the three types of component subproblems and each resulting type is described below. This subsection then concludes with comments about combining a group of component subproblems into a decomposition subproblem.

Slippage Component Subproblems

Slippage component subproblems result from that portion of problem **LSD LEFT** [E.6] that involve the primal slippage variables and constraints at nodes in periods $t = 1, \ldots, \bar{t}$. Let $\Omega_S^{(\bar{t})}\left(\mathbf{X}_{\bar{t}}^{(0)}\right)$, referred to as the slippage value function, denote the problem formed by this portion:

$$\Omega_S^{(\bar{t})}\left(\mathbf{X}_{\bar{t}}^{(0)}\right) = \max \sum_{t=1}^{\bar{t}} \sum_{h_t=1}^{H_t} \mathbf{c}'^{(1)[\bullet]_t} \mathbf{x}^{(1)[\bullet]_t}$$

$$\text{s.t. slippage constraints at nodes } [\bullet]_t = (t, h_t)\,,\ h_t = 1, \ldots, H_t,\ t = 1, \ldots, \bar{t},$$

$$\begin{aligned} \mathbf{W}^{(1,1)[\bullet]_t} \mathbf{x}^{(1)[\bullet]_t} &\leq \mathbf{b}^{(1)[\bullet]_t} - \mathbf{W}^{(1,0)[\bullet]_t} \mathbf{x}^{(0)[\bullet]_t} - \mathbf{B}^{(1,0)[\bullet]_t} \mathbf{x}^{(0)[\bullet]_{t-1}}, \\ \mathbf{x}^{(1)[\bullet]_t} &\geq \mathbf{0}. \end{aligned}$$

The slippage value function in scalar notation is then:

$$\Omega_{\mathrm{S}}^{(\bar{t})}\left(\mathbf{X}_{\bar{t}}^{(0)}\right) = \max -\sum_{t=1}^{\bar{t}} \hat{\varrho}_t \left[\sum_{h_t=1}^{H_t} \hat{p}^{[\bullet]_t} \left(\sum_{n=1}^{N}\sum_{k=1}^{K} \Phi_{kn} X_{kn}^{(1)[\bullet]_t}\right)\right] \qquad \text{[E.7]}$$

s.t. slippage constraints at nodes $[\bullet]_t = (t, h_t)\,,\ h_t = 1, \ldots, H_t,\ t = 1, \ldots, \bar{t},$

$$\begin{aligned}
-\sum_{k=1}^{K} X_{kn}^{(1)[\bullet]_t} &\leq \left(\triangle Y_{nt}^{\min} + x_n^{(0)[\bullet]_{t-1}} - x_n^{(0)[\bullet]_t}\right), \quad n = 1, \ldots, N,\\
-\sum_{k=1}^{K} X_{kn}^{(1)[\bullet]_t} &\leq -\left(\triangle Y_{nt}^{\min} + x_n^{(0)[\bullet]_{t-1}} - x_n^{(0)[\bullet]_t}\right), \quad n = 1, \ldots, N,\\
X_{kn}^{(1)[\bullet]_t} &\leq \triangle \Psi_{kn}, \quad k = 1, \ldots, K-1, \quad n = 1, \ldots, N,\\
X_{kn}^{(1)[\bullet]_t} &\geq 0, \qquad k = 1, \ldots, K, \qquad n = 1, \ldots, N.
\end{aligned}$$

Problem [E.7] is separable into independent subproblems for each security at each node in periods $t = 1, \ldots, \bar{t}$. These subproblems are the slippage component subproblems and are denoted by $\Omega_{\mathrm{S}n}^{[\bullet]_t}\left(\mathbf{X}_{\bar{t}}^{(0)}\right)$. Then, for all securities $n = 1, \ldots, N$ at all nodes $[\bullet]_t = (t, h_t)$, $h_t = 1, \ldots, H_t$, in periods $t = 1, \ldots, \bar{t}$:

$$\Omega_{\mathrm{S}n}^{[\bullet]_t}\left(\mathbf{X}_{\bar{t}}^{(0)}\right) = \max \ -\hat{\varrho}_t \hat{p}^{[\bullet]_t} \sum_{k=1}^{K} \Phi_{kn} X_{kn}^{(1)[\bullet]_t}$$

$$\begin{aligned}
\text{s.t.} \qquad -\sum_{k=1}^{K} X_{kn}^{(1)[\bullet]_t} &\leq \left(\triangle Y_{nt}^{\min} + x_n^{(0)[\bullet]_{t-1}} - x_n^{(0)[\bullet]_t}\right),\\
-\sum_{k=1}^{K} X_{kn}^{(1)[\bullet]_t} &\leq -\left(\triangle Y_{nt}^{\min} + x_n^{(0)[\bullet]_{t-1}} - x_n^{(0)[\bullet]_t}\right),\\
X_{kn}^{(1)[\bullet]_t} &\leq \triangle \Psi_{kn},\ k = 1, \ldots, K-1,\\
X_{kn}^{(1)[\bullet]_t} &\geq 0,\ k = 1, \ldots, K.
\end{aligned} \qquad \text{[E.8]}$$

Note that $\Omega_{Sn}^{[\bullet]_t}\left(\mathbf{X}_{\bar{t}}^{(0)}\right) \leq 0$ since each factor in the objective function coefficient for $X_{kn}^{(1)[\bullet]_t}$, $k = 1, \ldots, K$, is nonnegative. Therefore, zero is a valid upper bound for each $\theta_{Sj}^{[\bullet]_t}$ in the initial restricted master problem [E.2] for LSD. The dual to problem [E.8] is:

$$\Omega_{Sn}^{[\bullet]_t}\left(\mathbf{X}_{\bar{t}}^{(0)}\right) = \min \left(\triangle Y_{nt}^{\min} + x_n^{(0)[\bullet]_{t-1}} - x_n^{(0)[\bullet]_t}\right)\left(\pi_n^{(1)[\bullet]_t} - \pi_{N+n}^{(1)[\bullet]_t}\right) + \sum_{k=1}^{K-1} \triangle\Psi_{kn}\pi_{2N+(n-1)(K-1)+k}^{(1)[\bullet]_t} \qquad \text{[E.9]}$$

$$\begin{aligned}
\text{s.t.} \quad -\pi_n^{(1)[\bullet]_t} - \pi_{N+n}^{(1)[\bullet]_t} + \pi_{2N+(n-1)(K-1)+k}^{(1)[\bullet]_t} &\geq -\hat{\varrho}_t\hat{p}^{[\bullet]_t}\Phi_{kn}, \quad k = 1, \ldots, K-1, \\
-\pi_n^{(1)[\bullet]_t} - \pi_{N+n}^{(1)[\bullet]_t} &\geq -\hat{\varrho}_t\hat{p}^{[\bullet]_t}\Phi_{Kn}, \\
\pi_n^{(1)[\bullet]_t} &\geq 0, \\
\pi_n^{(1)[\bullet]_t} &\geq 0, \\
\pi_{2N+(n-1)(K-1)+k}^{(1)[\bullet]_t} &\geq 0, \qquad k = 1, \ldots, K-1.
\end{aligned}$$

Primal problems [E.8] are the slippage component subproblems for L-Shaped decomposition while dual problems [E.9] are the slippage component subproblems for Dantzig-Wolfe decomposition. Solutions to these subproblems are easily obtained by evaluating the primal and dual pair of subproblems for each security at each node in the applicable periods. Detailed solution procedures for problems [E.8] and [E.9] are given in Section E.2.

Deviation Component Subproblems

Deviation component subproblems result from that portion of problem **LSD LEFT** [E.6] that involve the primal deviation variables and constraints at nodes in periods $t = 1, \ldots, \bar{t}$. The primal deviation component subproblems, denoted by $\Omega_{D_{l_t}}^{[\bullet]_t}\left(\mathbf{X}_{\bar{t}}^{(0)}\right)$, are de-

rived in a manner similar to that for the slippage component subproblems. Then, for all outcomes $l_t = 1, \ldots, L_t$ at all nodes $[\bullet]_t = (t, h_t)$, $h_t = 1, \ldots, H_t$, in periods $t = 1, \ldots, \bar{t}$:

$$\Omega_{\mathrm{D}l_t}^{[\bullet]_t}\left(\mathbf{X}_{\bar{t}}^{(0)}\right) = \max \quad -\hat{\varrho}_t \hat{p}^{[\bullet]_t} \lambda_t p_{l_t}^{[\bullet]_t} \sum_{q=1}^{Q} \gamma_q X_{ql_t}^{(2)[\bullet]_t}$$

$$\begin{aligned} \text{s.t.} \qquad -\sum_{q=1}^{Q} X_{ql_t}^{(2)[\bullet]_t} &\leq \hat{d}_{l_t}^{[\bullet]_t} - \sum_{n=1}^{N} \left(\mu_n^{[\bullet]_t} - R_{nl_t}^{[\bullet]_t}\right) x_n^{(0)[\bullet]_t}, \\ X_{ql_t}^{(2)[\bullet]_t} &\leq \triangle\varphi_q, \; q = 1, \ldots, Q-1, \\ X_{ql_t}^{(2)[\bullet]_t} &\geq 0, \; q = 1, \ldots, Q. \end{aligned} \qquad \text{[E.10]}$$

Note that $\Omega_{\mathrm{D}l_t}^{[\bullet]_t}\left(\mathbf{X}_{\bar{t}}^{(0)}\right) \leq 0$ since each factor in the objective function coefficient for $X_{ql_t}^{(2)[\bullet]_t}$, $k = 1, \ldots, K$, is nonnegative. Therefore, zero is a valid upper bound for each $\theta_{\mathrm{D}j}^{[\bullet]_t}$ in the initial restricted master problem [E.2] for LSD. The dual to problem [E.10] is:

$$\Omega_{\mathrm{D}l_t}^{[\bullet]_t}\left(\mathbf{X}_{\bar{t}}^{(0)}\right) = \min \left[\hat{d}_{l_t}^{[\bullet]_t} - \sum_{n=1}^{N} \left(\mu_n^{[\bullet]_t} - R_{nl_t}^{[\bullet]_t}\right) x_n^{(0)[\bullet]_t}\right] \pi_{l_t}^{(2)[\bullet]_t} + \sum_{q=1}^{Q-1} \triangle\varphi_q \pi_{L_t+(l_t-1)(Q-1)+q}^{(2)[\bullet]_t} \qquad \text{[E.11]}$$

$$\begin{aligned} \text{s.t.} \quad -\pi_{l_t}^{(2)[\bullet]_t} + \pi_{L_t+(l_t-1)(Q-1)+q}^{(2)[\bullet]_t} &\geq -\hat{\varrho}_t \hat{p}^{[\bullet]_t} \lambda_t p_{l_t}^{[\bullet]_t} \gamma_q, \quad q = 1, \ldots, Q-1, \\ -\pi_{l_t}^{(2)[\bullet]_t} &\geq -\hat{\varrho}_t \hat{p}^{[\bullet]_t} \lambda_t p_{l_t}^{[\bullet]_t} \gamma_Q, \\ \pi_{l_t}^{(2)[\bullet]_t} &\geq 0, \\ \pi_{L_t+(l_t-1)(Q-1)+q}^{(2)[\bullet]_t} &\geq 0, \qquad q = 1, \ldots, Q-1. \end{aligned}$$

Primal problems [E.10] are the deviation component subproblems for L-Shaped decomposition while dual problems [E.11] are the deviation component subproblems for Dantzig-Wolfe decomposition. Solutions to these subproblems are easily obtained by evaluating the primal and dual pair of subproblems for each outcome at each node in the applicable pe-

riods. Detailed solution procedures for problems [E.10] and [E.11] are given in Section E.3.

Nodal Component Subproblems

Nodal component subproblems result from that portion of problem **LSD LEFT** [E.6] that involve the primal variables and constraints at nodes in periods after the cutoff period, i.e., in periods $t = \hat{t}, \ldots, T$. Note that there are no nodal subproblems if $\hat{t} = \bar{t}+1 > T$. The primal nodal component subproblems, denoted by $\Omega^{(\hat{t})}_{\mathrm{N}h_{\hat{t}}}\left(\mathbf{x}^{(0)[\bullet]_{\bar{t}}}\right)$, at nodes $[\bullet]_{\hat{t}} = \left(\hat{t}, h_{\hat{t}}\right)$, $h_{\hat{t}} = 1, \ldots, H_{\hat{t}}$, are:

$$\begin{aligned}\Omega^{(\hat{t})}_{\mathrm{N}h_{\hat{t}}}\left(\mathbf{x}^{(0)[\bullet]_{\bar{t}}}\right) = \max\ & \hat{\varrho}_{\hat{t}}\hat{p}^{[\bullet]_{\hat{t}}}\left(\sum_{n=1}^{N}\mu_n^{[\bullet]_{\hat{t}}}x_n^{(0)[\bullet]_{\hat{t}}} - \sum_{n=1}^{N}\sum_{k=1}^{K}\Phi_{kn}X_{kn}^{(1)[\bullet]_{\hat{t}}}\right.\\ & \left.-\lambda_{\hat{t}}\sum_{l_{\hat{t}}=1}^{L_{\hat{t}}}p_{l_{\hat{t}}}^{[\bullet]_{\hat{t}}}\sum_{q=1}^{Q}\gamma_q X_{ql_{\hat{t}}}^{(2)[\bullet]_{\hat{t}}}\right)\\ & + \sum_{t=\hat{t}+1}^{T}\hat{\varrho}_t\left[\sum_{h_t=1}^{H_t}\hat{p}^{[\bullet]_t}\left(\sum_{n=1}^{N}\mu_n^{[\bullet]_t}x_n^{(0)[\bullet]_t}\right.\right.\\ & \left.\left.-\sum_{n=1}^{N}\sum_{k=1}^{K}\Phi_{kn}X_{kn}^{(1)[\bullet]_t} - \lambda_t\sum_{l_t=1}^{L_t}p_{l_t}^{[\bullet]_t}\sum_{q=1}^{Q}\gamma_q X_{ql_t}^{(2)[\bullet]_t}\right)\right]\end{aligned} \qquad \text{[E.12]}$$

s.t. constraints at node $[\bullet]_{\hat{t}} = \left(\hat{t}, h_{\hat{t}}\right)$,

$$\begin{aligned}
\sum_{n=1}^{N}\sigma_n^{[\bullet]_{\hat{t}}}x_n^{(0)[\bullet]_{\hat{t}}} &\leq \hat{\beta}^{[\bullet]_{\hat{t}}}, \\
x_n^{(0)[\bullet]_{\hat{t}}} - \sum_{k=1}^{K}X_{kn}^{(1)[\bullet]_{\hat{t}}} &\leq \left(\Delta Y_{nt}^{\min} + x_n^{(0)[\bullet]_{\bar{t}}}\right), \quad n = 1, \ldots, N, \\
-x_n^{(0)[\bullet]_{\hat{t}}} - \sum_{k=1}^{K}X_{kn}^{(1)[\bullet]_{\hat{t}}} &\leq -\left(\Delta Y_{nt}^{\min} + x_n^{(0)[\bullet]_{\bar{t}}}\right), \quad n = 1, \ldots, N, \\
\sum_{n=1}^{N}\left(\mu_n^{[\bullet]_{\hat{t}}} - R_{nl_{\hat{t}}}^{[\bullet]_{\hat{t}}}\right)x_n^{(0)[\bullet]_{\hat{t}}} - \sum_{q=1}^{Q}X_{ql_t}^{(2)[\bullet]_{\hat{t}}} &\leq \hat{d}_{l_{\hat{t}}}^{[\bullet]_{\hat{t}}}, \qquad l_{\hat{t}} = 1, \ldots, L_{\hat{t}},
\end{aligned}$$

constraints at nodes $[\bullet]_t = (t, h_t)$, $h_t = 1, \ldots, H_t$, $t = \hat{t}+1, \ldots, T$,

$$
\begin{aligned}
\sum_{n=1}^{N} \sigma_n^{[\bullet]_t} x_n^{(0)[\bullet]_t} &\leq \hat{\beta}^{[\bullet]_t}, \\
-x_n^{(0)[\bullet]_{t-1}} + x_n^{(0)[\bullet]_t} - \sum_{k=1}^{K} X_{kn}^{(1)[\bullet]_t} &\leq \triangle Y_{nt}^{\min}, \quad n = 1, \ldots, N, \\
x_n^{(0)[\bullet]_{t-1}} - x_n^{(0)[\bullet]_t} - \sum_{k=1}^{K} X_{kn}^{(1)[\bullet]_t} &\leq -\triangle Y_{nt}^{\min}, \quad n = 1, \ldots, N, \\
\sum_{n=1}^{N} \left(\mu_n^{[\bullet]_t} - R_{nl_t}^{[\bullet]_t} \right) x_n^{(0)[\bullet]_t} - \sum_{q=1}^{Q} X_{ql_t}^{(2)[\bullet]_t} &\leq \hat{d}_{l_t}^{[\bullet]_t}, \quad l_t = 1, \ldots, L_t,
\end{aligned}
$$

and bounds at nodes $[\bullet]_{\hat{t}} = (\hat{t}, h_{\hat{t}})$ and $[\bullet]_t = (t, h_t)$, $h_t = 1, \ldots, H_t$, $t = \hat{t}+1, \ldots, T$,

$$
\begin{aligned}
&X_{kn}^{(1)[\bullet]_t} \leq \triangle \Psi_{kn}, \quad k = 1, \ldots, K-1, \quad n = 1, \ldots, N, \\
&X_{ql_t}^{(2)[\bullet]_t} \leq \triangle \varphi_q, \quad q = 1, \ldots, Q-1, \quad l_t = 1, \ldots, L_t, \\
&x_n^{(0)[\bullet]_t} \geq 0, \quad n = 1, \ldots, N, \\
&X_{kn}^{(1)[\bullet]_t} \geq 0, \quad k = 1, \ldots, K, \quad n = 1, \ldots, N, \\
&X_{ql_t}^{(2)[\bullet]_t} \geq 0, \quad q = 1, \ldots, Q, \quad l_t = 1, \ldots, L_t.
\end{aligned}
$$

The dual to problem [E.12] is:

$$
\begin{aligned}
\Omega_{Nh_{\hat{t}}}^{(\hat{t})} \left(\mathbf{x}^{(0)[\bullet]_{\bar{t}}} \right) = \min \Bigg[& \hat{\beta}^{[\bullet]_{\hat{t}}} \pi_0^{(0)[\bullet]_{\hat{t}}} + \sum_{n=1}^{N} \left(\triangle Y_{nt}^{\min} + x_n^{(0)[\bullet]_{\bar{t}}} \right) \left(\pi_n^{(1)[\bullet]_{\hat{t}}} - \pi_{N+n}^{(1)[\bullet]_{\hat{t}}} \right) \\
& + \sum_{n=1}^{N} \sum_{k=1}^{K-1} \triangle \Psi_{kn} \pi_{2N+(n-1)(K-1)+k}^{(1)[\bullet]_{\hat{t}}} \\
& + \sum_{l_{\hat{t}}=1}^{L_{\hat{t}}} \hat{d}_{l_{\hat{t}}}^{[\bullet]_{\hat{t}}} \pi_{l_{\hat{t}}}^{(2)[\bullet]_{\hat{t}}} + \sum_{l_{\hat{t}}=1}^{L_{\hat{t}}} \sum_{q=1}^{Q-1} \triangle \varphi_q \pi_{L_{\hat{t}}+\left(l_{\hat{t}}-1\right)(Q-1)+q}^{(2)[\bullet]_{\hat{t}}} \Bigg] \\
& + \sum_{t=\hat{t}+1}^{T} \sum_{h_t=1}^{H_t} \Bigg[\hat{\beta}^{[\bullet]_t} \pi_0^{(0)[\bullet]_t} + \sum_{n=1}^{N} \triangle Y_{nt}^{\min} \left(\pi_n^{(1)[\bullet]_t} - \pi_{N+n}^{(1)[\bullet]_t} \right) \\
& \quad + \sum_{n=1}^{N} \sum_{k=1}^{K-1} \triangle \Psi_{kn} \pi_{2N+(n-1)(K-1)+k}^{(1)[\bullet]_t} \\
& \quad + \sum_{l_t=1}^{L_t} \hat{d}_{l_t}^{[\bullet]_t} \pi_{l_t}^{(2)[\bullet]_t} + \sum_{l_t=1}^{L_t} \sum_{q=1}^{Q-1} \triangle \varphi_q \pi_{L_t+(l_t-1)(Q-1)+q}^{(2)[\bullet]_t} \Bigg]
\end{aligned}
\qquad \text{[E.13]}
$$

s.t. constraints at nodes $[\bullet]_{\hat{t}} = (\hat{t}, h_{\hat{t}})$ and $[\bullet]_t = (t, h_t)$, $h_t = 1, \ldots, H_t$, $t = \hat{t}+1, \ldots, T$,

$$\sigma_n^{[\bullet]_t}\pi_0^{(0)[\bullet]_t} + \pi_n^{(1)[\bullet]_t} - \pi_{N+n}^{(1)[\bullet]_t} + \sum_{l_t=1}^{L_t}\left(\mu_n^{[\bullet]_t} - R_{nl_t}^{[\bullet]_t}\right)\pi_{l_t}^{(2)[\bullet]_t}$$

$$- \sum_{l_t=1}^{L_t}\left(\pi_n^{(1)[\bullet]_{t+1}} - \pi_{N+n}^{(1)[\bullet]_{t+1}}\right) \geq \hat{\varrho}_t\hat{p}^{[\bullet]_t}\mu_n^{[\bullet]_t}, \quad n = 1,\ldots,N,$$

$$-\pi_n^{(1)[\bullet]_t} - \pi_{N+n}^{(1)[\bullet]_t} + \pi_{2N+(n-1)(K-1)+k}^{(1)[\bullet]_t} \geq -\hat{\varrho}_t\hat{p}^{[\bullet]_t}\Phi_{kn}, \qquad \begin{matrix} k = 1,\ldots,K-1, \\ n = 1,\ldots,N, \end{matrix}$$

$$-\pi_n^{(1)[\bullet]_t} - \pi_{N+n}^{(1)[\bullet]_t} \geq -\hat{\varrho}_t\hat{p}^{[\bullet]_t}\Phi_{Kn}, \qquad n = 1,\ldots,N,$$

$$-\pi_{l_t}^{(2)[\bullet]_t} + \pi_{L_t+(l_t-1)(Q-1)+q}^{(2)[\bullet]_t} \geq -\hat{\varrho}_t\hat{p}^{[\bullet]_t}\lambda_t p_{l_t}^{[\bullet]_t}\gamma_q, \qquad \begin{matrix} q = 1,\ldots,Q-1, \\ l_t = 1,\ldots,L_t, \end{matrix}$$

$$-\pi_{l_t}^{(2)[\bullet]_t} \geq -\hat{\varrho}_t\hat{p}^{[\bullet]_t}\lambda_t p_{l_t}^{[\bullet]_t}\gamma_Q, \qquad l_t = 1,\ldots,L_t,$$

and lower bounds at nodes $[\bullet]_{\hat{t}} = \left(\hat{t}, h_{\hat{t}}\right)$ and $[\bullet]_t = (t, h_t)$, $h_t = 1,\ldots,H_t, t = \hat{t}+1,\ldots,T$,

$$\pi_0^{(0)[\bullet]_t} \geq 0,$$

$$\pi_j^{(1)[\bullet]_t} \geq 0, \quad j = 1,\ldots,(K+1)N,$$

$$\pi_j^{(2)[\bullet]_t} \geq 0, \quad j = 1,\ldots,QL_t.$$

Primal problems [E.12] are the nodal component subproblems for L-Shaped decomposition while dual problems [E.13] are the nodal component subproblems for Dantzig-Wolfe decomposition. Solutions to these subproblems are obtained using a simplex solver.

Procedures for obtaining an upper bound, say $\hat{\Omega}_{Nh_{\hat{t}}}^{(\hat{t})}$, for each nodal component subproblem in period $\hat{t}$ are given in Section E.4. Each upper bound is independent of $\mathbf{x}^{(0)[\bullet]_{\bar{t}}}$ and can be determined prior to the first decomposition iteration. These upper bounds can then be used to establish upper bounds on the designated decomposition subproblems. For example, assume that the decomposition strategies call for a decomposition subproblem at each node in period $\hat{t}$. There are $H_{\hat{t}}$ nodal decomposition subproblems with a one-to-one

correspondence to the nodal component subproblems. The associated relaxation variables are then bounded by:

$$\theta_{\mathrm{N}h_{\hat{t}}}^{(\hat{t})} \leq \theta_{\mathrm{N}h_{\hat{t}}}^{(\hat{t})\max} = \hat{\Omega}_{\mathrm{N}h_{\hat{t}}}^{(\hat{t})}, h_{\hat{t}} = 1, \ldots, H_{\hat{t}}.$$

Upper bounds on the nodal component subproblems may also be combined to determine upper bounds on decomposition subproblems. Assume, for a second example, that the decomposition strategies designate one decomposition subproblem for all nodes in period $\hat{t}$. The associated relaxation variable is then bounded by:

$$\theta_{\mathrm{N1}}^{(\hat{t})} \leq \theta_{\mathrm{N1}}^{(\hat{t})\max} = \sum_{h_{\hat{t}}=1}^{H_{\hat{t}}} \hat{\Omega}_{\mathrm{N}h_{\hat{t}}}^{(\hat{t})}.$$

Combining Component Subproblems Into Decomposition Subproblems

Solution values, say $\mathbf{X}_{\bar{t}}^{(0)(G)}$, are obtained by solving the RMP at each iteration G of the decomposition algorithm. These are the values of the RMP primal variables in L-Shaped decomposition or they are the values of the dual multipliers to the non-convexity constraints in Dantzig-Wolfe decomposition. Solutions for decomposition subproblems are then obtained by combining solutions to the appropriate component subproblems described above.

For instance, assume that the decomposition strategies call for a subproblem accounting for all slippage at each node in periods $t = 1, \ldots, \bar{t}$. Solutions to slippage component subproblems $\Omega_{\mathrm{S}n}^{[\bullet]_t}\left(\mathbf{X}_{\bar{t}}^{(0)(G)}\right)$, $n = 1, \ldots, N$, would then be combined for each node in the applicable periods to obtain the solutions for the decomposition subproblems. The re-

sulting solutions to the decomposition subproblems are used to generate a new slippage optimality cut (LSD) or extreme point activity (DWD) corresponding to each node in periods 1 through $\bar{t}$. Let $\pi^{(1)[\bullet]_{\tilde{t}}(G)}$ be the vector of solutions to the slippage dual component subproblems [E.9] at some node $[\bullet]_{\tilde{t}} = (\tilde{t}, h_{\tilde{t}})$ in period $\tilde{t} \in \{1, \ldots, \bar{t}\}$ and denote the corresponding relaxation variable by $\theta_S^{[\bullet]_{\tilde{t}}}$. A new optimality cut for LSD would then have the form:

$$\left[\sum_{n=1}^{N}\left(\pi_n^{(1)[\bullet]_{\tilde{t}}(G)} - \pi_{N+n}^{(1)[\bullet]_{\tilde{t}}(G)}\right)\left(-x_n^{(1)[\bullet]_{\tilde{t}-1}} + x_n^{(1)[\bullet]_{\tilde{t}}}\right)\right] + \theta_S^{[\bullet]_{\tilde{t}}} \leq \left[\sum_{n=1}^{N}\left(\Delta Y_{nt}^{\min} + \sum_{k=1}^{K-1}\Psi_{kn}\pi_{2N+(n-1)(K-1)+k}^{(1)[\bullet]_{\tilde{t}}(G)}\right)\right] \quad \text{[E.14]}$$

A new extreme point activity for DWD would have an objective function coefficient equal to the right-hand-side of inequality [E.14] and technology matrix coefficients equal to the coefficients on the left-hand-side of the inequality.

Numerous decomposition strategies can be devised that take advantage of the flexibility offered by component subproblems [E.8] through [E.13]. Specific examples of decomposition strategies are given in Chapter 6.

E.1.3 Single-Period DWD/LSD

Model **MIMPSLP** problems with a single period have a cutoff period of $\bar{t} = T = 1$. These problems can be solved with a grand LP formulation or with decomposition using the procedures described for two-stage DWD/LSD in the next subsection. Decomposition procedures treat the primal slippage and deviation variables and constraints as second stage

components. Grand LP solution procedures may be more efficient (faster) than decomposition procedures when there are a few thousand or less random outcomes to consider.

Single period problems also occur in problems with multiple periods when the decomposition major strategy dictates that $\bar{t} = T - 1$. In this case, there is a single period problem anchored at each node in the terminal period. This can be an effective strategy when there are many nodes (hundreds or more) in the final period.

E.1.4 Two-Stage DWD/LSD

Two-stage Dantzig-Wolfe or L-Shaped decomposition can be applied to any model **MIMPSLP** problem regardless of the number of periods assigned to the problem. The block-separable recourse property of model problems provides for significant flexibility in structuring problems for decomposition.

One obvious strategy is to assign a cutoff period of $\bar{t} = T$ so that the first stage of LSD consists of all primal budget constraints while all primal slippage and deviation constraints are accounted for in second stage subproblems. The comparable DWD procedure would have all dual composite constraints in the first stage while all dual slippage and deviation constraints would be in second stage subproblems. All subproblems in this strategy can be solved by combining solutions of the easily solved slippage and deviation component subproblems.

A second strategy is to assign the cutoff period such that $\bar{t} < T$. The resulting decomposition procedure would then involve subproblems from all three categories: slippage,

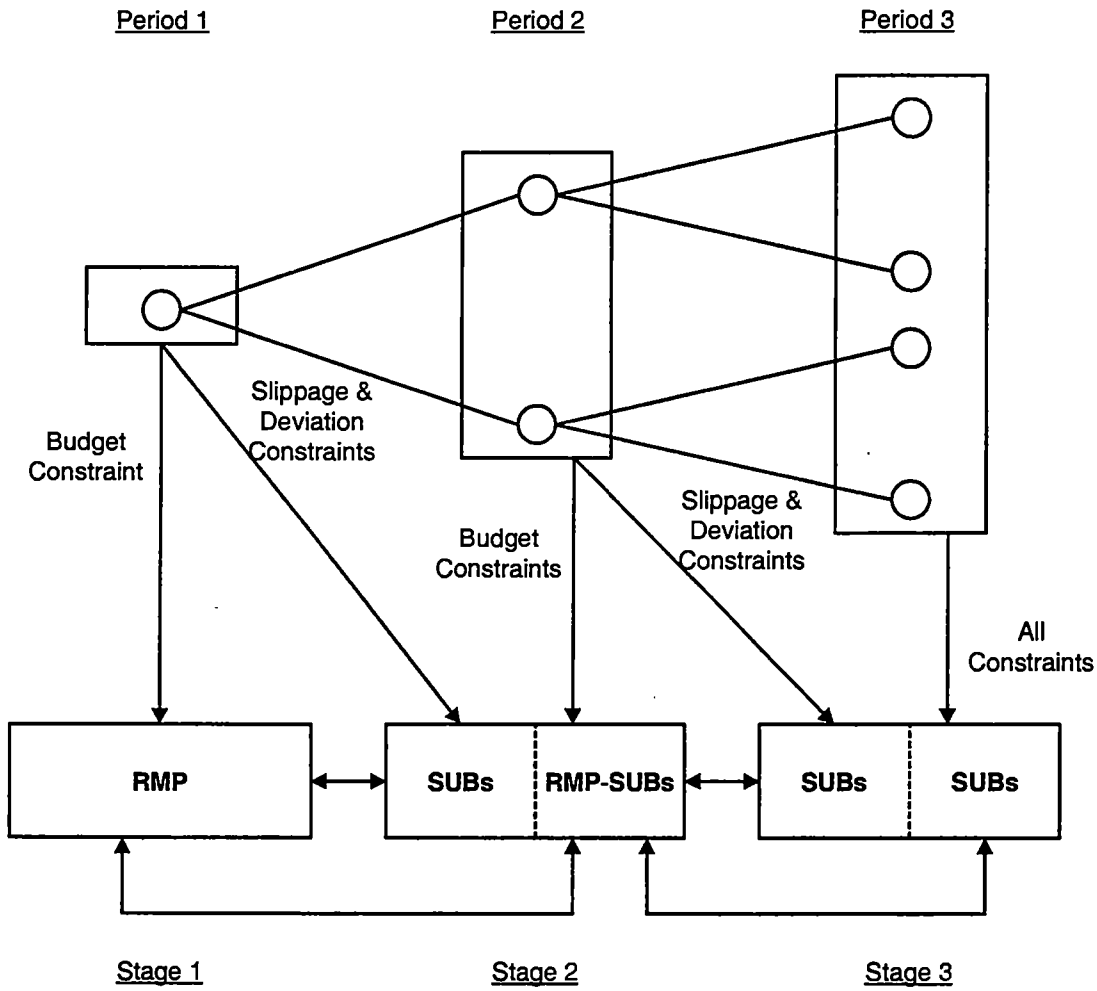
deviation, and nodal. Initial RMP formulations would be smaller than those with $\bar{t} = T$ since only budget/composite constraints in the first $\bar{t}$ periods are present. Second stage subproblems would account for the slippage and deviation variables and constraints in periods $t = 1, \ldots, \bar{t}$ as well as all variables and constraints in periods after the cutoff period. Note that two-stage decomposition can only be applied when grand LP solutions are obtained for the nodal component subproblems anchored at the nodes in the period following the cutoff period, i.e., in period $\hat{t} = \bar{t} + 1$. Multi-stage, or nested, decomposition described in Section E.1.5 below is required when DWD or LSD is applied to the nodal component subproblems.

Algorithms **LSD(multicut)** and **DWD(multiactivities)** described in Sections 3.1.4 and 3.2.3 respectively can be applied, with minor modifications, to the resulting two-stage problems. Each algorithm can be modified for model **MIMPSLP** problems by replacing the initialization procedures described in the referenced sections with much simpler and significantly faster procedures. The detailed calculations of the documented initialization procedures are not required since model problems are bounded and have complete recourse. Procedure **LSD(multicut)-Initialize** can be replaced by a procedure that simply sets the bounds on the relaxation variables as illustrated in the formulations of problem **LSD RMP**(1) [E.1] (array notation) and [E.2] (scalar notation). Similarly, procedure **DWD(multiactivities)-Initialize** can be replaced by a procedure that sets the objective function coefficients for the convexity variables as illustrated in the formulations of problem **DWD RMP**(1) [E.4] (array notation) and [E.5] (scalar notation).

E.1.5 Multi-Stage DWD/LSD

Multi-stage, or nested, decomposition procedures described in Section 4.2 starting on page 117 can be applied to model **MIMPSLP** problems with multiple periods. The nested decomposition strategies described in Section 4.4.2 can be employed directly if the block-separable recourse property of model problems is ignored. Strategies described in Section 4.4.4 are applicable, however, if the block-separable recourse property is utilized.

Block-separable recourse allows for increased flexibility in structuring problems for decomposition. All master and subproblems would resemble the nodal component subproblems described in Section E.1.2 if block-separability is ignored. Employment of the block-separable recourse property, however, also yields subproblems that may be solved using the slippage and deviation component subproblems also described in Section E.1.2. For example, Figure E.1 illustrates nested L-Shaped decomposition applied to a three-period model problem when block-separable recourse is utilized. Each of the three periods coincides with a stage in a three-stage decomposition strategy. Primal budget variables and constraints in each period remain in the corresponding stage and are accounted for in the RMP in the first stage, the RMP-SUB problems in stage two, and the SUB problems of stage three. Primal slippage and deviation variables and constraints in the first two periods, on the other hand, are used to form subproblems in the second and third stages. The latter subproblems are solved easily and efficiently by combining solutions of the slippage and deviation component subproblems as appropriate. Specific nested decomposition



NOTES

1. Three-period problem solved with nested L-Shaped Decomposition in three stages.
2. First period budget constraint forms the initial first stage RMP.
3. First period slippage and deviation constraints form second stage subproblems that receive solution information from and send cuts to the first stage RMP.
4. Budget constraints at second period nodes form initial second stage RMP-SUBs. SUB mode receives solution information from and sends cuts to the first stage. RMP mode sends solution information to and receives cuts from the third stage.
5. Second period slippage and deviation constraints form third stage subproblems that receive solution information from and send cuts to the parent second stage RMP-SUB.
6. Data at each third period node is used to create a third stage single-period subproblem that receives solution information from and sends cuts to the parent second stage RMP-SUB.
7. Subproblem results in each stage may be used to create individual cuts or aggregated to form composite cuts.

Figure E.1: Example Nested Decomposition of a **MIMPSLP** Three-Period Problem

strategies that use the block-separable recourse property of model **MIMPSLP** problems are described in the Chapter 6.

E.1.6 Myopic Decomposition

The myopic decomposition procedures described in Section 4.5 can be directly applied to model **MIMPSLP** problems since these problems have complete recourse. Primal and dual subproblems are formulated according to the descriptions given in Section 4.5.1. The myopic primal subproblem, corresponding to problem [4.18] on page 154, for the single first period node is:

$$\mathfrak{M}_{\mathrm{P}}^{[]}(\,) = \max \sum_{n=1}^{N} \mu_n^{[]} x_n^{(0)[]} - \sum_{n=1}^{N}\sum_{k=1}^{K} \Phi_{kn} X_{kn}^{(1)[]} - \lambda_1 \sum_{l_1=1}^{L_1} p_{l_1}^{[]} \sum_{q=1}^{Q} \gamma_q X_{ql_1}^{(2)[]} \qquad \text{[E.15]}$$

$$\text{s.t.} \quad \sum_{n=1}^{N} \sigma_n^{[]} x_n^{(0)[]} \leq \hat{\beta}^{[]},$$

$$x_n^{(0)[]} - \sum_{k=1}^{K} X_{kn}^{(1)[]} \leq \Delta Y_{n1}^{\min}, \quad n = 1, \ldots, N,$$

$$-x_n^{(0)[]} - \sum_{k=1}^{K} X_{kn}^{(1)[]} \leq -\Delta Y_{n1}^{\min}, \quad n = 1, \ldots, N,$$

$$\sum_{n=1}^{N} \left(\mu_n^{[]} - R_{nl_1}^{[]}\right) x_n^{(0)[]} - \sum_{q=1}^{Q} X_{ql_1}^{(2)[]} \leq \hat{d}_{l_1}^{[]}, \quad l_1 = 1, \ldots, L_1,$$

$$X_{kn}^{(1)[]} \leq \Delta\Psi_{kn}, \quad k = 1, \ldots, K-1, \quad n = 1, \ldots, N,$$

$$X_{ql_1}^{(2)[]} \leq \Delta\varphi_q, \quad q = 1, \ldots, Q-1, \quad l_1 = 1, \ldots, L_1,$$

$$x_n^{(0)[]} \geq 0, \quad n = 1, \ldots, N,$$

$$X_{kn}^{(1)[]} \geq 0, \quad k = 1, \ldots, K, \quad n = 1, \ldots, N,$$

$$X_{ql_1}^{(2)[]} \geq 0, \quad q = 1, \ldots, Q, \quad l_1 = 1, \ldots, L_1.$$

Appendix E Decomposition of Model MIMPSLP Problems

Myopic primal subproblems, corresponding to problems [4.19] on page 154, for nodes $[\bullet]_t = (t, h_t)$, $h_t = 1, \ldots, H_t$, in periods $t = 2, \ldots, T$ are:

$$\mathfrak{M}_{\mathrm{P}}^{[\bullet]_t}\left(\mathbf{x}^{[\bullet]_{t-1}}\right) = \max\ \hat{\varrho}_t \hat{p}^{[\bullet]_t} \left(\sum_{n=1}^{N} \mu_n^{[\bullet]_t} x_n^{(0)[\bullet]_t} - \sum_{n=1}^{N} \sum_{k=1}^{K} \Phi_{kn} X_{kn}^{(1)[\bullet]_t} \right. \qquad [\mathrm{E}.16]$$
$$\left. -\lambda_t \sum_{l_t=1}^{L_t} p_{l_t}^{[\bullet]_t} \sum_{q=1}^{Q} \gamma_q X_{ql_t}^{(2)[\bullet]_t} \right)$$

$$\text{s.t.} \quad \sum_{n=1}^{N} \sigma_n^{[\bullet]_t} x_n^{(0)[\bullet]_t} \leq \hat{\beta}^{[\bullet]_t},$$

$$x_n^{(0)[\bullet]_t} - \sum_{k=1}^{K} X_{kn}^{(1)[\bullet]_t} \leq \left(\triangle Y_{nt}^{\min} + x_n^{(0)[\bullet]_{t-1}} \right), \quad n = 1, \ldots, N,$$

$$-x_n^{(0)[\bullet]_t} - \sum_{k=1}^{K} X_{kn}^{(1)[\bullet]_t} \leq -\left(\triangle Y_{nt}^{\min} + x_n^{(0)[\bullet]_{t-1}} \right), \quad n = 1, \ldots, N,$$

$$\sum_{n=1}^{N} \left(\mu_n^{[\bullet]_t} - R_{nl_t}^{[\bullet]_t} \right) x_n^{(0)[\bullet]_t} - \sum_{q=1}^{Q} X_{ql_t}^{(2)[\bullet]_t} \leq \hat{d}_{l_t}^{[\bullet]_t}, \quad l_t = 1, \ldots, L_t,$$

$$X_{kn}^{(1)[\bullet]_t} \leq \triangle \Psi_{kn}, \quad k = 1, \ldots, K-1, \quad n = 1, \ldots, N,$$

$$X_{ql_t}^{(2)[\bullet]_t} \leq \triangle \varphi_q, \quad q = 1, \ldots, Q-1, \quad l_t = 1, \ldots, L_t,$$

$$x_n^{(0)[\bullet]_t} \geq 0, \quad n = 1, \ldots, N,$$

$$X_{kn}^{(1)[\bullet]_t} \geq 0, \quad k = 1, \ldots, K, \quad n = 1, \ldots, N,$$

$$X_{ql_t}^{(2)[\bullet]_t} \geq 0, \quad q = 1, \ldots, Q, \quad l_t = 1, \ldots, L_t.$$

Appendix E Decomposition of Model MIMPSLP Problems

The myopic dual subproblems, corresponding to problems [4.20] on page 154, for nodes $[\bullet]_T = (T, h_T)$, $h_T = 1, \ldots, H_T$, in the terminal period are:

$$\mathfrak{M}_{\mathrm{D}}^{[\bullet]_T}(\,) = \min\ \hat{\beta}^{[\bullet]_T}\pi_0^{(0)[\bullet]_T} + \sum_{n=1}^{N}\triangle Y_{nT}^{\min}\left(\pi_n^{(1)[\bullet]_T} - \pi_{N+n}^{(1)[\bullet]_T}\right) + \sum_{n=1}^{N}\sum_{k=1}^{K-1}\triangle\Psi_{kn}\pi_{2N+(n-1)(K-1)+k}^{(1)[\bullet]_T} + \sum_{l_T=1}^{L_T}\hat{d}_{l_T}^{[\bullet]_T}\pi_{l_T}^{(2)[\bullet]_T} + \sum_{l_T=1}^{L_T}\sum_{q=1}^{Q-1}\triangle\varphi_q\pi_{L_T+(l_T-1)(Q-1)+q}^{(2)[\bullet]_T} \qquad [\mathrm{E.17}]$$

$$\text{s.t.}\quad \sigma_n^{[\bullet]_T}\pi_0^{(0)[\bullet]_T} + \pi_n^{(1)[\bullet]_T} - \pi_{N+n}^{(1)[\bullet]_T} + \sum_{l_T=1}^{L_T}\left(\mu_n^{[\bullet]_T} - R_{nl_T}^{[\bullet]_T}\right)\pi_{l_T}^{(2)[\bullet]_T} \geq \hat{\varrho}_T\hat{p}^{[\bullet]_T}\mu_n^{[\bullet]_T},\quad n = 1, \ldots, N,$$

$$\begin{aligned}
&-\pi_n^{(1)[\bullet]_T} - \pi_{N+n}^{(1)[\bullet]_T} + \pi_{2N+(n-1)(K-1)+k}^{(1)[\bullet]_T} \geq -\hat{\varrho}_T\hat{p}^{[\bullet]_T}\Phi_{kn}, && k = 1, \ldots, K-1,\ n = 1, \ldots, N,\\
&-\pi_n^{(1)[\bullet]_T} - \pi_{N+n}^{(1)[\bullet]_T} \geq -\hat{\varrho}_T\hat{p}^{[\bullet]_T}\Phi_{Kn}, && n = 1, \ldots, N,\\
&-\pi_{l_T}^{(2)[\bullet]_T} + \pi_{L_T+(l_T-1)(Q-1)+q}^{(2)[\bullet]_T} \geq -\hat{\varrho}_T\hat{p}^{[\bullet]_T}\lambda_T p_{l_T}^{[\bullet]_T}\gamma_q, && q = 1, \ldots, Q-1,\ l_T = 1, \ldots, L_T,\\
&-\pi_{l_T}^{(2)[\bullet]_T} \geq -\hat{\varrho}_T\hat{p}^{[\bullet]_T}\lambda_T p_{l_T}^{[\bullet]_T}\gamma_Q, && l_T = 1, \ldots, L_T,
\end{aligned}$$

$$\begin{aligned}
&\pi_0^{(0)[\bullet]_T} \geq 0,\\
&\pi_j^{(1)[\bullet]_T} \geq 0,\quad j = 1, \ldots, (K+1)\,N,\\
&\pi_j^{(2)[\bullet]_T} \geq 0,\quad j = 1, \ldots, QL_T.
\end{aligned}$$

Appendix E Decomposition of Model MIMPSLP Problems

Myopic dual subproblems, corresponding to problems [4.21] and [4.22], for nodes $[\bullet]_t = (t, h_t)$, $h_t = 1, \ldots, H_t$, in periods $t = T-1, \ldots, 1$ are:

$$\mathfrak{M}_{\mathrm{D}}^{[\bullet]_t}\left(\left[\pi^{[\bullet]_{t+1}}\right]_{l_t=1}^{L_t}\right) = \min\ \hat{\beta}^{[\bullet]_t}\pi_0^{(0)[\bullet]_t} \qquad \text{[E.18]}$$
$$+\sum_{n=1}^{N}\triangle Y_{nt}^{\min}\left(\pi_n^{(1)[\bullet]_t}-\pi_{N+n}^{(1)[\bullet]_t}\right)+\sum_{n=1}^{N}\sum_{k=1}^{K-1}\triangle\Psi_{kn}\pi_{2N+(n-1)(K-1)+k}^{(1)[\bullet]_t}$$
$$+\sum_{l_t=1}^{L_T}\hat{d}_{l_t}^{[\bullet]_t}\pi_{l_t}^{(2)[\bullet]_t}+\sum_{l_t=1}^{L_t}\sum_{q=1}^{Q-1}\triangle\varphi_q\pi_{L_t+(l_t-1)(Q-1)+q}^{(2)[\bullet]_t}$$

$$\text{s.t.}\quad \sigma_n^{[\bullet]_t}\pi_0^{(0)[\bullet]_t}+\pi_n^{(1)[\bullet]_t}-\pi_{N+n}^{(1)[\bullet]_t}+\sum_{l_t=1}^{L_t}\left(\mu_n^{[\bullet]_t}-R_{nl_t}^{[\bullet]_t}\right)\pi_{l_t}^{(2)[\bullet]_t}$$
$$\geq\left[\hat{\varrho}_t\hat{p}^{[\bullet]_t}\mu_n^{[\bullet]_t}+\sum_{l_t=1}^{L_t}\left(\pi_n^{(1)[\bullet]_{t+1}}-\pi_{N+n}^{(1)[\bullet]_{t+1}}\right)\right],\quad n=1,\ldots,N,$$

$$-\pi_n^{(1)[\bullet]_t}-\pi_{N+n}^{(1)[\bullet]_t}+\pi_{2N+(n-1)(K-1)+k}^{(1)[\bullet]_t}\geq-\hat{\varrho}_t\hat{p}^{[\bullet]_t}\Phi_{kn},\qquad \begin{matrix}k=1,\ldots,K-1,\\ n=1,\ldots,N,\end{matrix}$$

$$-\pi_n^{(1)[\bullet]_t}-\pi_{N+n}^{(1)[\bullet]_t}\geq-\hat{\varrho}_t\hat{p}^{[\bullet]_t}\Phi_{Kn},\qquad n=1,\ldots,N,$$

$$-\pi_{l_t}^{(2)[\bullet]_t}+\pi_{L_t+(l_t-1)(Q-1)+q}^{(2)[\bullet]_t}\geq-\hat{\varrho}_t\hat{p}^{[\bullet]_t}\lambda_t p_{l_t}^{[\bullet]_t}\gamma_q,\qquad \begin{matrix}q=1,\ldots,Q-1,\\ l_t=1,\ldots,L_t,\end{matrix}$$

$$-\pi_{l_t}^{(2)[\bullet]_t}\geq-\hat{\varrho}_t\hat{p}^{[\bullet]_t}\lambda_t p_{l_t}^{[\bullet]_t}\gamma_Q,\qquad l_t=1,\ldots,L_t,$$

$$\pi_0^{(0)[\bullet]_t}\geq 0,$$

$$\pi_j^{(1)[\bullet]_t}\geq 0,\quad j=1,\ldots,(K+1)N,$$

$$\pi_j^{(2)[\bullet]_t}\geq 0,\quad j=1,\ldots,QL_t.$$

E.2 Solving Slippage Component Subproblems

The solution to the primal slippage component subproblem [E.8] on page 371 is determined first. Dual problem [E.9] on page 372 is then solved using the primal solution and duality theory. The derivations rely upon the fact that each factor in the objective function coefficients of the primal problem [E.8] are nonnegative:

$$\begin{aligned} \hat{\varrho}_t &> 0, \quad t = 1, \ldots, T, \\ \hat{p}^{[\bullet]_t} &> 0, \quad h_t = 1, \ldots, H_t, \quad t = 1, \ldots, T, \\ \Phi_{kn} &\geq 0, \quad k = 1, \ldots, K, \quad n = 1, \ldots, N. \end{aligned}$$

Define Γ such that:

$$\Gamma = \triangle Y_{nt}^{\min} + x_n^{(0)[\bullet]_{t-1}} - x_n^{(0)[\bullet]_t}, \qquad \text{[E.19]}$$

then the first two constraints of the primal slippage component subproblem [E.8] imply that:

$$\sum_{k=1}^{K} X_{kn}^{(1)[\bullet]_t} = |\Gamma| . \qquad \text{[E.20]}$$

The primal upper bounding constraints on $X_{kn}^{(1)[\bullet]_t}$ insure that for any $\bar{k} \in \{1, \ldots, K-1\}$:

$$\sum_{k=1}^{\bar{k}} X_{kn}^{(1)[\bullet]_t} \leq \sum_{k=1}^{\bar{k}} \triangle \Psi_{kn}$$

and since $\triangle \Psi_{kn} = \Psi_{kn} - \Psi_{(k-1)n}$ for $k = 1, \ldots, K$ (see the third row from the bottom of Table 5.2 on page 185) with $\Psi_{0n} = 0$:

$$\sum_{k=1}^{\bar{k}} X_{kn}^{(1)[\bullet]_t} \leq \sum_{k=1}^{\bar{k}} \left(\Psi_{kn} - \Psi_{(k-1)n} \right) = \Psi_{\bar{k}n}. \qquad \text{[E.21]}$$

Inequalities [5.4b],

$$\Phi_{1n} < \cdots < \Phi_{kn} < \cdots < \Phi_{Kn},$$

and the objective function and upper bounding constraints on $X_{kn}^{(1)[\bullet]_t}$ of problem [E.8] imply that:

$$X_{kn}^{(1)[\bullet]_t} > 0 \text{ only if } X_{(k-1)n}^{(1)[\bullet]_t} = \triangle\Psi_{(k-1)n} \text{ for any } k \in \{2, \ldots, K\}. \qquad \text{[E.22]}$$

Define $\bar{k}$ such that:

$$\bar{k} = \max k \in \{0, \ldots, K-1\} \text{ with } \Psi_{kn} \leq |\Gamma|, \qquad \text{[E.23]}$$

which determines the largest Ψ_{kn} that is less than or equal to $|\Gamma|$ since Ψ_{kn} increases monotonically with index k by relationships [5.4a]:

$$0 = \Psi_{0n} < \Psi_{1n} < \cdots < \Psi_{kn} < \cdots < \Psi_{Kn} = \infty,$$

so that:

$$\Psi_{\bar{k}n} \leq |\Gamma| \text{ and } \Psi_{(\bar{k}+1)n} > |\Gamma|.$$

Relations [E.19] through [E.23] then imply the primal solution is:

$$X_{kn}^{(1)[\bullet]_t} = \begin{cases} \Psi_{kn} - \Psi_{(k-1)n} = \triangle\Psi_{kn}, & k = 1, \ldots, \bar{k}, \\ |\Gamma| - \Psi_{\bar{k}n}, & k = \bar{k}+1, \\ 0, & k = \bar{k}+2, \ldots, K. \end{cases} \qquad \text{[E.24]}$$

The solution to the dual slippage component subproblem [E.9] is solved by examining the problem under two cases defined by the relationship between $\Psi_{\bar{k}n}$ and Γ.

Case 1: $\Psi_{\bar{k}n} < |\Gamma|$

Note that $\Psi_{\bar{k}n} < |\Gamma|$ implies that $|\Gamma| > 0$ due to relationships [5.4a]:

$$0 = \Psi_{0n} < \Psi_{1n} < \cdots < \Psi_{kn} < \cdots < \Psi_{Kn} = \infty.$$

Furthermore, $\Psi_{(\bar{k}+1)n} > |\Gamma|$ by the definition [E.23] of $\bar{k}$. Complementary slackness and primal solution [E.24] then require that

$$\pi^{(1)[\bullet]_t}_{2N+(n-1)(K-1)+k} = 0, k = \bar{k}+1, \ldots, K-1,$$

since the complementary primal constraints

$$X^{(1)[\bullet]_t}_{kn} \leq \Delta\Psi_{kn}, k = \bar{k}+1, \ldots, K-1,$$

are not tight. Complementary slackness also requires that the first $(\bar{k}+1)$ constraints of dual problem [E.9] be tight since the complementary primal variables are positive by primal solution [E.24] and the case condition, $\Psi_{\bar{k}n} < |\Gamma|$. In addition, the objective function of the dual problem implies that only one of $\pi^{(1)[\bullet]_t}_n$ and $\pi^{(1)[\bullet]_t}_{N+n}$ can be positive in order to minimize the function value. Therefore, the dual solution under Case 1 is:

$$\pi^{(1)[\bullet]_t}_n = \begin{cases} \hat{\varrho}_t \hat{p}^{[\bullet]_t} \Phi_{(\bar{k}+1)n}, & \text{if } \Gamma < 0, \\ 0, & \text{if } \Gamma > 0, \end{cases}$$

$$\pi^{(1)[\bullet]_t}_{N+n} = \begin{cases} 0, & \text{if } \Gamma < 0, \\ \hat{\varrho}_t \hat{p}^{[\bullet]_t} \Phi_{(\bar{k}+1)n}, & \text{if } \Gamma > 0, \end{cases} \qquad \text{[E.25]}$$

$$\pi^{(1)[\bullet]_t}_{2N+(n-1)(K-1)+k} = \begin{cases} \hat{\varrho}_t \hat{p}^{[\bullet]_t} \left(\Phi_{(\bar{k}+1)n} - \Phi_{kn} \right), & k = 1, \ldots, \bar{k}, \\ 0, & k = \bar{k}+1, \ldots, K-1. \end{cases}$$

Case 2: $\Psi_{\bar{k}n} = |\Gamma|$

Complementary slackness, primal solution [E.24], and the case condition, $\Psi_{\bar{k}n} = |\Gamma|$, require, as in Case 1, that:

$$\pi^{(1)[\bullet]_t}_{2N+(n-1)(K-1)+k} = 0, k = \bar{k}+1, \ldots, K-1,$$

since the complementary primal constraints

$$X_{kn}^{(1)[\bullet]_t} \leq \triangle\Psi_{kn},\ k = \bar{k}+1,\ldots,K-1,$$

are not tight. Complementary slackness also requires that the first $\bar{k}$ constraints of dual problem [E.9] be tight since the complementary primal variables are positive by primal solution [E.24] and the case condition, implying that:

$$-\pi_n^{(1)[\bullet]_t} - \pi_{N+n}^{(1)[\bullet]_t} + \pi_{2N+(n-1)(K-1)+k}^{(1)[\bullet]_t} = -\hat{\varrho}_t\hat{p}^{[\bullet]_t}\Phi_{kn},\ k = 1,\ldots,\bar{k}. \qquad \text{[E.26]}$$

Strong duality requires that the objective functions of problems [E.8] and [E.9] be equal at optimality, which, with equations [E.19] and [E.24], indicates that:

$$\Gamma\left(\pi_n^{(1)[\bullet]_t} - \pi_{N+n}^{(1)[\bullet]_t}\right) + \sum_{k=1}^{\bar{k}} \triangle\Psi_{kn}\pi_{2N+(n-1)(K-1)+k}^{(1)[\bullet]_t} = -\hat{\varrho}_t\hat{p}^{[\bullet]_t}\sum_{k=1}^{\bar{k}}\Phi_{kn}\triangle\Psi_{kn}. \qquad \text{[E.27]}$$

Equations [E.26] and [E.27] appear to be a system of $\left(\bar{k}+1\right)$ equations in $\left(\bar{k}+2\right)$ unknowns. The objective function of the dual problem [E.9] implies, however, the further condition:

$$\begin{aligned} \pi_n^{(1)[\bullet]_t} \geq 0 \text{ and } \pi_{N+n}^{(1)[\bullet]_t} = 0, &\quad \text{if } \Gamma \leq 0, \\ \pi_n^{(1)[\bullet]_t} = 0 \text{ and } \pi_{N+n}^{(1)[\bullet]_t} \geq 0, &\quad \text{if } \Gamma > 0, \end{aligned} \qquad \text{[E.28]}$$

in order to minimize the function value. There are, therefore, a balanced number of equations and unknowns. Multiplying each equation k at [E.26] by $\triangle\Psi_{kn}$ and summing the results yields:

$$-\left(\pi_n^{(1)[\bullet]_t} + \pi_{N+n}^{(1)[\bullet]_t}\right)\sum_{k=1}^{\bar{k}}\triangle\Psi_{kn} + \sum_{k=1}^{\bar{k}}\triangle\Psi_{kn}\pi_{2N+(n-1)(K-1)+k}^{(1)[\bullet]_t} = -\hat{\varrho}_t\hat{p}^{[\bullet]_t}\sum_{k=1}^{\bar{k}}\Phi_{kn}\triangle\Psi_{kn}. \qquad \text{[E.29]}$$

Since,

$$\sum_{k=1}^{\bar{k}} \triangle\Psi_{kn} = \sum_{k=1}^{\bar{k}} \left(\Psi_{kn} - \Psi_{(k-1)n}\right) = \Psi_{\bar{k}n},$$

equation [E.29] becomes,

$$-\left(\pi_n^{(1)[\bullet]_t} + \pi_{N+n}^{(1)[\bullet]_t}\right)\Psi_{\bar{k}n} + \sum_{k=1}^{\bar{k}} \triangle\Psi_{kn}\pi_{2N+(n-1)(K-1)+k}^{(1)[\bullet]_t} = -\hat{\varrho}_t\hat{p}^{[\bullet]_t}\sum_{k=1}^{\bar{k}}\Phi_{kn}\triangle\Psi_{kn}. \qquad \text{[E.30]}$$

The case condition, $\Psi_{\bar{k}n} = |\Gamma|$, and the additional condition [E.28] imply that:

$$-\left(\pi_n^{(1)[\bullet]_t} + \pi_{N+n}^{(1)[\bullet]_t}\right)\Psi_{\bar{k}n} = \Gamma\left(\pi_n^{(1)[\bullet]_t} - \pi_{N+n}^{(1)[\bullet]_t}\right),$$

so that equation [E.30] can be written as:

$$\Gamma\left(\pi_n^{(1)[\bullet]_t} - \pi_{N+n}^{(1)[\bullet]_t}\right) + \sum_{k=1}^{\bar{k}} \triangle\Psi_{kn}\pi_{2N+(n-1)(K-1)+k}^{(1)[\bullet]_t} = -\hat{\varrho}_t\hat{p}^{[\bullet]_t}\sum_{k=1}^{\bar{k}}\Phi_{kn}\triangle\Psi_{kn},$$

which is equation [E.27]. Therefore, equations [E.26] and [E.27] are not linearly independent. The coefficient matrix formed by omitting equation [E.27] and the first two columns of equations [E.26] (since at least one of $\pi_n^{(1)[\bullet]_t}$ and $\pi_{N+n}^{(1)[\bullet]_t}$ is zero by condition [E.28] while the other is omitted) is the non-singular $\bar{k}$-by-$\bar{k}$ identity matrix. One of the $\left(\bar{k}+2\right)$ variables:

$$\pi_n^{(1)[\bullet]_t} \text{ or } \pi_{N+n}^{(1)[\bullet]_t} \text{ or } \pi_{2N+(n-1)(K-1)+k}^{(1)[\bullet]_t},\ k \in \left\{1,\ldots,\bar{k}\right\},$$

may then be assigned an arbitrary nonnegative value and the other values will follow. Consequently, dual problem [E.9] has an infinite number of alternate optimal solutions under Case 2. Solution values under Case 2 are assigned to maintain consistency with Case 1.

Therefore, solution set [E.25] is selected if $\Gamma \neq 0$, otherwise,

$$\begin{aligned} \pi_n^{(1)[\bullet]_t} &= \hat{\varrho}_t \hat{p}^{[\bullet]_t} \Phi_{1n}, \\ \pi_{N+n}^{(1)[\bullet]_t} &= 0, \\ \pi_{2N+(n-1)(K-1)+k}^{(1)[\bullet]_t} &= 0, k = 1, \ldots, K-1, \end{aligned} \qquad \text{[E.31]}$$

when $\Gamma = 0$ which implies that $\bar{k} = 0$ by the definition [E.23] of $\bar{k}$.

E.3 Solving Deviation Component Subproblems

The solution to the primal deviation component subproblem [E.10] on page 373 is determined first. Dual problem [E.11] on page 373 is then solved using the primal solution and duality theory. The derivations rely upon the fact that each factor in the objective function coefficients of the primal problem [E.10] are nonnegative:

$$\begin{aligned} \hat{\varrho}_t &> 0, \quad t = 1, \ldots, T, \\ \hat{p}^{[\bullet]_t} &> 0, \quad h_t = 1, \ldots, H_t, \quad t = 1, \ldots, T, \\ \lambda_t &> 0, \quad t = 1, \ldots, T, \\ p_{l_t}^{[\bullet]_t} &> 0, \quad l_t = 1, \ldots, L_t, \quad h_t = 1, \ldots, H_t, \quad t = 1, \ldots, T, \\ \gamma_q &\geq 0, \quad q = 1, \ldots, Q. \end{aligned}$$

Define Γ such that:

$$\Gamma = \hat{d}_{l_t}^{[\bullet]_t} - \sum_{n=1}^{N} \left(\mu_n^{[\bullet]_t} - R_{nl_t}^{[\bullet]_t} \right) x_n^{(0)[\bullet]_t}. \qquad \text{[E.32]}$$

Then if $\Gamma \geq 0$, the optimal solution to the primal deviation component subproblem [E.10] is clearly $\mathbf{X}_{\bullet l_t}^{(2)[\bullet]_t} = \mathbf{0}$ with objective value $\Omega_{Dl_t}^{[\bullet]_t}\left(\mathbf{X}_{\bar{t}}^{(0)}\right) = 0$. Therefore, consider the case

when $\Gamma < 0$. The first constraint to the primal problem implies that:

$$-\sum_{q=1}^{Q} X_{ql_t}^{(2)[\bullet]_t} \leq \Gamma,$$

$$\Longrightarrow \sum_{q=1}^{Q} X_{ql_t}^{(2)[\bullet]_t} \geq |\Gamma|. \qquad \text{[E.33]}$$

The primal upper bounding constraints on $X_{ql_t}^{(2)[\bullet]_t}$ insure that for any $\bar{q} \in \{1, \ldots, Q-1\}$:

$$\sum_{q=1}^{\bar{q}} X_{ql_t}^{(2)[\bullet]_t} \leq \sum_{q=1}^{\bar{q}} \triangle\varphi_q$$

and since $\triangle\varphi_q = \varphi_q - \varphi_{q-1}$ for $q = 1, \ldots, Q$ (see the second row from the bottom of Table 5.2 on page 185) with $\varphi_0 = 0$:

$$\sum_{q=1}^{\bar{q}} X_{ql_t}^{(2)[\bullet]_t} \leq \sum_{q=1}^{\bar{q}} (\varphi_q - \varphi_{q-1}) = \varphi_{\bar{q}}. \qquad \text{[E.34]}$$

Inequalities [5.10b],

$$\gamma_1 < \cdots < \gamma_q < \cdots < \gamma_Q,$$

and the objective function and upper bounding constraints on $X_{ql_t}^{(2)[\bullet]_t}$ of problem [E.10] imply that:

$$X_{ql_t}^{(2)[\bullet]_t} > 0 \text{ only if } X_{(q-1)l_t}^{(2)[\bullet]_t} = \triangle\varphi_{q-1} \text{ for any } q \in \{2, \ldots, Q\}. \qquad \text{[E.35]}$$

Define $\bar{q}$ such that:

$$\bar{q} = \max q \in \{0, \ldots, Q-1\} \text{ with } \varphi_q \leq |\Gamma|, \qquad \text{[E.36]}$$

which determines the largest φ_q that is less than or equal to $|\Gamma|$ since φ_q increases monotonically with index q by relationships [5.10a]:

$$0 = \varphi_0 < \varphi_1 < \cdots < \varphi_q < \cdots < \varphi_Q = \infty,$$

so that:

$$\varphi_{\bar{q}} \leq |\Gamma| \text{ and } \varphi_{\bar{q}+1} > |\Gamma|.$$

Relations [E.32] through [E.36] and the objective function to problem [E.10] then imply the primal solution is:

$$\mathbf{X}_{\bullet l_t}^{(2)[\bullet]_t} = \mathbf{0} \text{ if } \Gamma \geq 0, \text{ otherwise, } X_{q l_t}^{(2)[\bullet]_t} = \begin{cases} \varphi_q - \varphi_{q-1} = \triangle\varphi_q, & q = 1, \ldots, \bar{q}, \\ |\Gamma| - \varphi_{\bar{q}}, & k = \bar{q}+1, \\ 0, & k = \bar{q}+2, \ldots, K. \end{cases} \quad \text{[E.37]}$$

The solution to the dual deviation component subproblem [E.11] is solved by examining the problem under three cases defined by the value of Γ and the relationship between $\varphi_{\bar{q}}$ and Γ.

Case 1: $\Gamma > 0$

Strong duality and primal solution [E.37] imply that the dual solution under Case 1 is:

$$\boldsymbol{\pi}^{(2)[\bullet]_t} = \mathbf{0}. \quad \text{[E.38]}$$

Case 2: $\varphi_{\bar{q}} < |\Gamma|, \Gamma < 0$

Note that $\varphi_{\bar{q}} < |\Gamma|$ implies that $|\Gamma| > 0$ due to relationships [5.10a]:

$$0 = \varphi_0 < \varphi_1 < \cdots < \varphi_q < \cdots < \varphi_Q = \infty.$$

Furthermore, $\varphi_{(\bar{q}+1)} > |\Gamma|$ by the definition [E.36] of $\bar{q}$. Complementary slackness and primal solution [E.37] then require that

$$\pi_{L_t+(l_t-1)(Q-1)+q}^{(2)[\bullet]_t} = 0, q = \bar{q}+1, \ldots, Q-1,$$

since the complementary primal constraints

$$X_{ql_t}^{(2)[\bullet]_t} \leq \Delta\varphi_q, q = \bar{q}+1, \ldots, Q-1,$$

are not tight. Complementary slackness also requires that the first $(\bar{q}+1)$ constraints of dual problem [E.11] be tight since the complementary primal variables are positive by primal solution [E.37] and the case condition, $\varphi_{\bar{q}} < |\Gamma|$. Therefore, the dual solution under Case 2 is:

$$\pi_{l_t}^{(2)[\bullet]_t} = \hat{\varrho}_t \hat{p}^{[\bullet]_t} \lambda_t p_{l_t}^{[\bullet]_t} \gamma_{\bar{q}+1} \qquad \text{[E.39]}$$

$$\pi_{L_t+(l_t-1)(Q-1)+q}^{(2)[\bullet]_t} = \begin{cases} \hat{\varrho}_t \hat{p}^{[\bullet]_t} \lambda_t p_{l_t}^{[\bullet]_t} \left(\gamma_{\bar{q}+1} - \varphi_q\right), & q = 1, \ldots, \bar{q}, \\ 0, & q = \bar{q}+1, \ldots, Q-1. \end{cases}$$

Case 3: $\varphi_{\bar{q}} = |\Gamma|, \Gamma \leq 0$

Complementary slackness, primal solution [E.37], and the case condition, $\varphi_{\bar{q}} = |\Gamma|$, require, as in Case 2, that:

$$\pi_{L_t+(l_t-1)(Q-1)+q}^{(2)[\bullet]_t} = 0, q = \bar{q}+1, \ldots, Q-1,$$

since the complementary primal constraints

$$X_{ql_t}^{(2)[\bullet]_t} \leq \Delta\varphi_q, q = \bar{q}+1, \ldots, Q-1,$$

are not tight. Complementary slackness also requires that the first $\bar{q}$ constraints of dual problem [E.11] be tight since the complementary primal variables are positive by primal solution [E.37] and the case condition, implying that:

$$-\pi_{l_t}^{(2)[\bullet]_t} + \pi_{L_t+(l_t-1)(Q-1)+q}^{(2)[\bullet]_t} = -\hat{\varrho}_t \hat{p}^{[\bullet]_t} \lambda_t p_{l_t}^{[\bullet]_t} \gamma_q, q = 1, \ldots, \bar{q}. \qquad \text{[E.40]}$$

Strong duality requires that the objective functions of problems [E.10] and [E.11] be equal at optimality, which, with equations [E.32] and [E.37], indicates that:

$$\Gamma\pi_{l_t}^{(2)[\bullet]_t} + \sum_{q=1}^{\bar{q}} \triangle\varphi_q \pi_{L_t+(l_t-1)(Q-1)+q}^{(2)[\bullet]_t} = -\hat{\varrho}_t \hat{p}^{[\bullet]_t} \lambda_t p_{l_t}^{[\bullet]_t} \sum_{q=1}^{\bar{q}} \gamma_q \triangle\varphi_q. \qquad \text{[E.41]}$$

Equations [E.40] and [E.41] define a system of $(\bar{q}+1)$ equations in $(\bar{q}+1)$ unknowns. Multiplying each equation q at [E.40] by $\triangle\varphi_q$ and summing the results yields:

$$-\pi_{l_t}^{(2)[\bullet]_t} \sum_{q=1}^{\bar{q}} \triangle\varphi_q + \sum_{q=1}^{\bar{q}} \triangle\varphi_q \pi_{L_t+(l_t-1)(Q-1)+q}^{(2)[\bullet]_t} = -\hat{\varrho}_t \hat{p}^{[\bullet]_t} \lambda_t p_{l_t}^{[\bullet]_t} \sum_{q=1}^{\bar{q}} \gamma_q \triangle\varphi_q. \qquad \text{[E.42]}$$

Since,

$$\sum_{q=1}^{\bar{q}} \triangle\varphi_q = \sum_{q=1}^{\bar{q}} \left(\varphi_q - \varphi_{q-1}\right) = \varphi_{\bar{q}},$$

equation [E.42] becomes,

$$-\pi_{l_t}^{(2)[\bullet]_t} \varphi_{\bar{q}} + \sum_{q=1}^{\bar{q}} \triangle\varphi_q \pi_{L_t+(l_t-1)(Q-1)+q}^{(2)[\bullet]_t} = -\hat{\varrho}_t \hat{p}^{[\bullet]_t} \lambda_t p_{l_t}^{[\bullet]_t} \sum_{q=1}^{\bar{q}} \gamma_q \triangle\varphi_q. \qquad \text{[E.43]}$$

The case condition, $\varphi_{\bar{q}} = |\Gamma|, \Gamma \leq 0$, then implies that:

$$\Gamma\pi_{l_t}^{(2)[\bullet]_t} + \sum_{q=1}^{\bar{q}} \triangle\varphi_q \pi_{L_t+(l_t-1)(Q-1)+q}^{(2)[\bullet]_t} = -\hat{\varrho}_t \hat{p}^{[\bullet]_t} \lambda_t p_{l_t}^{[\bullet]_t} \sum_{q=1}^{\bar{q}} \gamma_q \triangle\varphi_q,$$

which is equation [E.41]. Therefore, equations [E.40] and [E.41] are not linearly independent. The coefficient matrix formed by omitting equation [E.41] and the first column of equations [E.40] is the non-singular $\bar{q}$-by-$\bar{q}$ identity matrix. One of the $(\bar{q}+1)$ variables:

$$\pi_{l_t}^{(2)[\bullet]_t} \text{ or } \pi_{L_t+(l_t-1)(Q-1)+q}^{(2)[\bullet]_t},\ q \in \{1, \ldots, \bar{q}\},$$

may then be assigned an arbitrary nonnegative value and the other values will follow. Consequently, dual problem [E.11] has an infinite number of alternate optimal solutions under Case 3. Solution values under Case 3 are assigned to maintain consistency with Case 2.

Therefore, solution set [E.37] is selected if $\Gamma < 0$, otherwise,

$$\pi_{l_t}^{(2)[\bullet]_t} = \hat{\varrho}_t \hat{p}^{[\bullet]_t} \lambda_t p_{l_t}^{[\bullet]_t} \gamma_1, \qquad \text{[E.44]}$$

$$\pi_{L_t+(l_t-1)(Q-1)+q}^{(2)[\bullet]_t} = 0, q = 1, \ldots, Q-1,$$

when $\Gamma = 0$ which implies that $\bar{q} = 0$ by the definition [E.36] of $\bar{q}$.

E.4 Bounding Nodal Component Subproblems

Nodal component subproblems must be solved using a simplex algorithm. However, upper bounds on these subproblems can be used to place bounds on the LSD relaxation (θ) variables or to determine objective function coefficients for DWD convexity (η) variables. An upper bound on $\Omega_{Nh_{\hat{t}}}^{(\hat{t})}\left(\mathbf{x}^{(0)[\bullet]_{\bar{t}}}\right)$ is established by determining a feasible solution to the dual nodal component subproblem [E.13] on page 375. The derivations below rely upon the fact that each factor, except the mean returns $\mu_n^{[\bullet]_t}$, in the objective function coefficients of the primal problem [E.12] on page 374 are nonnegative:

$$\begin{aligned}
\hat{\varrho}_t &> 0, \quad t = 1, \ldots, T, \\
\hat{p}^{[\bullet]_t} &> 0, \quad h_t = 1, \ldots, H_t, \quad t = 1, \ldots, T, \\
\Phi_{kn} &\geq 0, \quad k = 1, \ldots, K, \quad n = 1, \ldots, N, \\
\lambda_t &> 0, \quad t = 1, \ldots, T, \\
p_{l_t}^{[\bullet]_t} &> 0, \quad l_t = 1, \ldots, L_t, \quad h_t = 1, \ldots, H_t, \quad t = 1, \ldots, T, \\
\gamma_q &\geq 0, \quad q = 1, \ldots, Q.
\end{aligned}$$

and that all standard deviations $\sigma_n^{[\bullet]_t}$ are positive.

Note that

$$\left.\begin{array}{l}\boldsymbol{\pi}^{(1)[\bullet]_t}=\mathbf{0},\\ \boldsymbol{\pi}^{(2)[\bullet]_t}=\mathbf{0},\end{array}\right\}[\bullet]_t=\left(\hat{t},h_{\hat{t}}\right)\text{ and }[\bullet]_t=\left(t,h_t\right),h_t=1,\ldots,H_t,t=\hat{t}+1,\ldots,T,$$

[E.45]

are feasible to all constraints in the dual problem [E.13]. A feasible solution to the dual problem can then be defined by determining a value for $\pi_0^{(0)[\bullet]_t}$ at each node represented in the problem such that $\pi_0^{(0)[\bullet]_t}$, $\boldsymbol{\pi}^{(1)[\bullet]_t}=\mathbf{0}$, and $\boldsymbol{\pi}^{(2)[\bullet]_t}=\mathbf{0}$ are feasible to the sets of dual composite constraints:

$$\sigma_n^{[\bullet]_t}\pi_0^{(0)[\bullet]_t}+\pi_n^{(1)[\bullet]_t}-\pi_{N+n}^{(1)[\bullet]_t}+\sum_{l_t=1}^{L_t}\left(\mu_n^{[\bullet]_t}-R_{nl_t}^{[\bullet]_t}\right)\pi_{l_t}^{(2)[\bullet]_t}$$
$$-\sum_{l_t=1}^{L_t}\left(\pi_n^{(1)[\bullet]_{t+1}}-\pi_{N+n}^{(1)[\bullet]_{t+1}}\right)\geq\hat{\varrho}_t\hat{p}^{[\bullet]_t}\mu_n^{[\bullet]_t},\quad n=1,\ldots,N,\quad\text{[E.46]}$$

at these nodes.

Define $\Gamma_n^{[\bullet]_t}$ such that:

$$\Gamma_n^{[\bullet]_t}=\begin{cases}\frac{\hat{\varrho}_t\hat{p}^{[\bullet]_t}\mu_n^{[\bullet]_t}}{\sigma_n^{[\bullet]_t}}, & \text{if } \mu_n^{[\bullet]_t}>0,\\ 0, & \text{if } \mu_n^{[\bullet]_t}\leq 0,\end{cases}\quad\text{for}\quad\begin{cases}n=1,\ldots,N\text{ at nodes}\\ [\bullet]_t=\left(\hat{t},h_{\hat{t}}\right)\text{ and}\\ [\bullet]_t=\left(t,h_t\right),h_t=1,\ldots,H_t,t=\hat{t}+1,\ldots,T,\end{cases}$$

and $n^{[\bullet]_t}$ such that:

$$n^{[\bullet]_t}=\left\{n\in\{1,\ldots,N\}\;\middle|\;\Gamma_n^{[\bullet]_t}\geq\Gamma_j^{[\bullet]_t},j=1,\ldots,N\right\}$$
$$\text{at nodes}\begin{cases}[\bullet]_t=\left(\hat{t},h_{\hat{t}}\right), & \text{and}\\ [\bullet]_t=\left(t,h_t\right), & h_t=1,\ldots,H_t,\\ & t=\hat{t}+1,\ldots,T.\end{cases}$$

Then,

$$\pi_0^{(0)[\bullet]_t}=\Gamma_{n^{[\bullet]_t}}^{[\bullet]_t},\ [\bullet]_t=\left(\hat{t},h_{\hat{t}}\right)\text{ and }[\bullet]_t=\left(t,h_t\right),h_t=1,\ldots,H_t,t=\hat{t}+1,\ldots,T,$$

[E.47]

and $\pi^{(1)[\bullet]_t}$, $\pi^{(2)[\bullet]_t}$ as defined by equations [E.45] are feasible to the dual composite constraints [E.46] and thus to the dual nodal component subproblem [E.13].

Therefore, with $\pi_0^{(0)[\bullet]_{\hat{t}}}$ determined by equations [E.47],

$$\hat{\Omega}_{Nh_{\hat{t}}}^{(\hat{t})} = \hat{\beta}^{[\bullet]_{\hat{t}}}\pi_0^{(0)[\bullet]_{\hat{t}}} + \sum_{t=\hat{t}+1}^{T}\sum_{h_t=1}^{H_t}\hat{\beta}^{[\bullet]_t}\pi_0^{(0)[\bullet]_t},\ h_{\hat{t}} = 1,\ldots,H_{\hat{t}}, \qquad \text{[E.48]}$$

establish upper bounds on each nodal component subproblem since the dual variable values defined by equations [E.45] and [E.47] are feasible but not necessarily optimum to problem [E.13]. Note that the upper bounds defined by equations [E.48] are independent of $\mathbf{x}^{(0)[\bullet]_{\bar{t}}}$ and can therefore be determined prior to the first decomposition iteration.

Appendix F
Expected Mahalanobis Squared Distances

Proofs for Propositions 13 through 15 in Section 5.6.4 are given below. These propositions establish the expected Mahalanobis squared distance (MSD) values used in model **MIMPSLP**. Notational conventions and preliminary results required by the proofs are developed first followed by the three proofs.

F.1 Notation and Preliminary Results

Inductive reasoning will be used in all three proofs below. Each proposition will be proven valid for a M-variate distribution given validity for a $(M-1)$-variate distribution where $M \geq 2$. Therefore, let a $(M-1)$-variate distribution be described by mean vector $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_{M-1})'$ and positive definite covariance matrix $\boldsymbol{\Sigma}_{M-1} \in \mathbb{R}^{(M-1)\times(M-1)}$. Let $\mathbf{r} = (r_1, \ldots, r_{M-1})'$ represent a random vector from the distribution and denote the covariances in matrix $\boldsymbol{\Sigma}_{M-1}$ by σ_{ij}, i.e.,

$$\sigma_{ij} = \mathcal{E}\left[(r_i - \mu_i)\,(r_j - \mu_j)\right], i = 1, \ldots, M-1, j = i = 1, \ldots, M-1,$$

with

$$\sigma_{ii} = \mathcal{E}\left[(r_i - \mu_i)^2\right] = \sigma_i^2, i = 1, \ldots, M-1,$$

where $\mathcal{E}$ is the expectation operator.

Appendix F Expected Mahalanobis Squared Distances

Let the notation $\langle \mathbf{v} \rangle_M$ represent the vector of the first M elements of vector $\mathbf{v}$ when the dimension of $\mathbf{v}$ is greater than or equal to M. Assume a new variable r_M with mean μ_M and covariances $\boldsymbol{\sigma}_M = \left(\sigma_{1M}, \ldots, \sigma_{(M-1)M}, \sigma_M^2\right)'$ is added to the distribution such that the new covariance matrix

$$\boldsymbol{\Sigma}_M = \begin{bmatrix} \boldsymbol{\Sigma}_{M-1} & \langle \boldsymbol{\sigma}_M \rangle_{M-1} \\ \langle \boldsymbol{\sigma}_M \rangle'_{M-1} & \sigma_M^2 \end{bmatrix} \qquad \text{[F.1]}$$

is positive definite. Noble and Daniel [159, problem 13 on page 39] indicate that the inverse matrix to $\boldsymbol{\Sigma}_M$ is:

$$\boldsymbol{\Sigma}_M^{-1} = \begin{bmatrix} \mathbf{A}_{M-1} & \langle \mathbf{a}_M \rangle_{M-1} \\ \langle \mathbf{a}_M \rangle'_{M-1} & a_{MM} \end{bmatrix}, \qquad \text{[F.2]}$$

where $\mathbf{A}_{M-1} \in \mathbb{R}^{(M-1)\times(M-1)}$ and $\mathbf{a}_M = (a_{1M}, \ldots, a_{MM})'$ such that:

$$a_{MM} = \frac{1}{\sigma_M^2 - \langle \boldsymbol{\sigma}_M \rangle'_{M-1} \boldsymbol{\Sigma}_{M-1}^{-1} \langle \boldsymbol{\sigma}_M \rangle_{M-1}}, \qquad \text{[F.3a]}$$

$$\langle \mathbf{a}_M \rangle_{M-1} = -a_{MM} \boldsymbol{\Sigma}_{M-1}^{-1} \langle \boldsymbol{\sigma}_M \rangle_{M-1}, \qquad \text{[F.3b]}$$

$$\mathbf{A}_{M-1} = \boldsymbol{\Sigma}_{M-1}^{-1} + \frac{1}{a_{MM}} \langle \mathbf{a}_M \rangle_{M-1} \langle \mathbf{a}_M \rangle'_{M-1}. \qquad \text{[F.3c]}$$

The MSD, $\mathcal{M}_{\boldsymbol{\Sigma}_M}(\mathbf{v}, \mathbf{w})$, for two vectors $\mathbf{v} \in \mathbb{R}^M$ and $\mathbf{w} \in \mathbb{R}^M$ associated with the new M-variate distribution can then be written in terms of $\mathcal{M}_{\boldsymbol{\Sigma}_{M-1}}\left(\langle \mathbf{v} \rangle_{M-1}, \langle \mathbf{w} \rangle_{M-1}\right)$ and three constant terms. First, the definition of MSD established by equation [5.57] on page 216 yields:

$$\mathcal{M}_{\boldsymbol{\Sigma}_M}(\mathbf{v}, \mathbf{w}) = (\mathbf{v} - \mathbf{w})' \boldsymbol{\Sigma}_M^{-1} (\mathbf{v} - \mathbf{w}).$$

Equation [F.2] then implies that:

$$\mathcal{M}_{\boldsymbol{\Sigma}_M}(\mathbf{v}, \mathbf{w}) = \begin{bmatrix} \langle \mathbf{v} - \mathbf{w} \rangle'_{M-1} & (v_M - w_M) \end{bmatrix} \begin{bmatrix} \mathbf{A}_{M-1} & \langle \mathbf{a}_M \rangle_{M-1} \\ \langle \mathbf{a}_M \rangle'_{M-1} & a_{MM} \end{bmatrix} \begin{bmatrix} \langle \mathbf{v} - \mathbf{w} \rangle_{M-1} \\ (v_M - w_M) \end{bmatrix}$$

which equates to:

$$\begin{aligned}\mathcal{M}_{\boldsymbol{\Sigma}_M}(\mathbf{v},\mathbf{w}) &= \langle \mathbf{v}-\mathbf{w}\rangle'_{M-1}\,\mathbf{A}_{M-1}\,\langle \mathbf{v}-\mathbf{w}\rangle_{M-1} + \langle \mathbf{v}-\mathbf{w}\rangle'_{M-1}\,\langle \mathbf{a}_M\rangle_{M-1}\,(v_M-w_M)\\ &\quad + (v_M-w_M)\,\langle \mathbf{a}_M\rangle'_{M-1}\,\langle \mathbf{v}-\mathbf{w}\rangle_{M-1} + (v_M-w_M)\,a_{MM}\,(v_M-w_M).\end{aligned}$$

Replacing $\mathbf{A}_{M-1}$ with the equivalent expression from equation [F.3c] and combining the middle two terms results in:

$$\begin{aligned}\mathcal{M}_{\boldsymbol{\Sigma}_M}(\mathbf{v},\mathbf{w}) &= \langle \mathbf{v}-\mathbf{w}\rangle'_{M-1}\,\boldsymbol{\Sigma}^{-1}_{M-1}\,\langle \mathbf{v}-\mathbf{w}\rangle_{M-1}\\ &\quad + \frac{1}{a_{MM}}\,\langle \mathbf{v}-\mathbf{w}\rangle'_{M-1}\,\langle \mathbf{a}_M\rangle_{M-1}\,\langle \mathbf{a}_M\rangle'_{M-1}\,\langle \mathbf{v}-\mathbf{w}\rangle_{M-1}\\ &\quad + 2\,(v_M-w_M)\,\langle \mathbf{a}_M\rangle'_{M-1}\,\langle \mathbf{v}-\mathbf{w}\rangle_{M-1} + a_{MM}\,(v_M-w_M)^2.\end{aligned}$$

The first term in the last equation above is equivalent to $\mathcal{M}_{\boldsymbol{\Sigma}_{M-1}}\left(\langle \mathbf{v}\rangle_{M-1}, \langle \mathbf{w}\rangle_{M-1}\right)$ implying that:

$$\begin{aligned}\mathcal{M}_{\boldsymbol{\Sigma}_M}(\mathbf{v},\mathbf{w}) &= \mathcal{M}_{\boldsymbol{\Sigma}_{M-1}}\left(\langle \mathbf{v}\rangle_{M-1}, \langle \mathbf{w}\rangle_{M-1}\right) + \frac{1}{a_{MM}}\left[\langle \mathbf{v}-\mathbf{w}\rangle'_{M-1}\,\langle \mathbf{a}_M\rangle_{M-1}\right]^2\\ &\quad + 2\,(v_M-w_M)\,\langle \mathbf{a}_M\rangle'_{M-1}\,\langle \mathbf{v}-\mathbf{w}\rangle_{M-1} + a_{MM}\,(v_M-w_M)^2.\end{aligned}$$

Finally, expanding the inner products in the middle two terms and squaring the results of the expansion of the second term yields:

$$\begin{aligned}\mathcal{M}_{\boldsymbol{\Sigma}_M}(\mathbf{v},\mathbf{w}) &= \mathcal{M}_{\boldsymbol{\Sigma}_{M-1}}\left(\langle \mathbf{v}\rangle_{M-1}, \langle \mathbf{w}\rangle_{M-1}\right) \qquad \text{[F.4]}\\ &\quad + \frac{1}{a_{MM}}\left[\sum_{m=1}^{M-1} a_{mM}^2\,(v_m-w_m)^2 + 2\sum_{i=1}^{M-2}\sum_{j=i+1}^{M-1} a_{iM}a_{jM}\,(v_i-w_i)\,(v_j-w_j)\right]\\ &\quad + 2\sum_{m=1}^{M-1} a_{mM}\,(v_m-w_m)\,(v_M-w_M) + a_{MM}\,(v_M-w_M)^2.\end{aligned}$$

Equation [F.4] is referenced by each of the three following proofs.

F.2 Proof of Proposition 13

Proposition 13 on page 218 is reproduced below. The proof follows the reproduction of the proposition.

Proposition 13 *Let* $\mathbf{v} \in \mathbb{R}^M$ *be a random vector from a* M*-variate distribution with mean vector* $\boldsymbol{\mu} \in \mathbb{R}^M$ *and positive definite covariance matrix* $\boldsymbol{\Sigma}_M \in \mathbb{R}^{M \times M}$*. Then,*

$$\mathcal{E}\left[\mathcal{M}_{\boldsymbol{\Sigma}_M}(\mathbf{v}, \boldsymbol{\mu})\right] = M.$$

Proof Proposition 13 is clearly true for $M = 1$ since

$$\mathcal{E}\left[\mathcal{M}_{\boldsymbol{\Sigma}_1}(v, \mu)\right] = \mathcal{E}\left[(v-\mu)' \boldsymbol{\Sigma}_1^{-1}(v-\mu)\right] = \frac{1}{\sigma^2}\mathcal{E}\left[(v-\mu)^2\right] = \frac{\sigma^2}{\sigma^2} = 1$$

where $\boldsymbol{\Sigma}_1 = [\sigma^2]$. Assume the proposition is true for $M-1$ where $M \geq 2$. Then, equation [F.4] with $\mathbf{w} = \boldsymbol{\mu}$ implies that:

$$\begin{aligned}
\mathcal{E}\left[\mathcal{M}_{\boldsymbol{\Sigma}_M}(\mathbf{v}, \boldsymbol{\mu})\right] &= \mathcal{E}\left[\mathcal{M}_{\boldsymbol{\Sigma}_{M-1}}\left(\langle \mathbf{v} \rangle_{M-1}, \langle \boldsymbol{\mu} \rangle_{M-1}\right)\right] \\
&\quad + \frac{1}{a_{MM}}\mathcal{E}\left[\sum_{m=1}^{M-1} a_{mM}^2 (v_m - \mu_m)^2 \right. \\
&\qquad \left. + 2\sum_{i=1}^{M-2}\sum_{j=i+1}^{M-1} a_{iM} a_{jM} (v_i - \mu_i)(v_j - \mu_j)\right] \\
&\quad + 2\mathcal{E}\left[\sum_{m=1}^{M-1} a_{mM}(v_m - \mu_m)(v_M - \mu_M)\right] + a_{MM}\mathcal{E}\left[(v_M - \mu_M)^2\right].
\end{aligned}$$

The induction assumption implies that $\mathcal{E}\left[\mathcal{M}_{\boldsymbol{\Sigma}_{M-1}}\left(\langle \mathbf{v}\rangle_{M-1}, \langle \boldsymbol{\mu}\rangle_{M-1}\right)\right] = M-1$ so that:

$$\begin{aligned}\mathcal{E}\left[\mathcal{M}_{\boldsymbol{\Sigma}_M}(\mathbf{v},\boldsymbol{\mu})\right] &= M-1+\frac{1}{a_{MM}}\left(\sum_{m=1}^{M-1} a_{mM}^2\sigma_m^2 + 2\sum_{i=1}^{M-2}\sum_{j=i+1}^{M-1} a_{iM}a_{jM}\sigma_{ij}\right) \\ &\quad +2\sum_{m=1}^{M-1} a_{mM}\sigma_{mM} + a_{MM}\sigma_M^2.\end{aligned}$$

Writing the first term on the second line of the last equation above as two summations and regrouping terms yields:

$$\begin{aligned}\mathcal{E}\left[\mathcal{M}_{\boldsymbol{\Sigma}_M}(\mathbf{v},\boldsymbol{\mu})\right] = M-1+&\left(\sum_{m=1}^{M-1} a_{mM}\sigma_{mM} + a_{MM}\sigma_M^2\right) \\ &+\sum_{m=1}^{M-1} a_{mM}\sigma_{mM} + \frac{1}{a_{MM}}\left(\sum_{m=1}^{M-1} a_{mM}^2\sigma_m^2 + 2\sum_{i=1}^{M-2}\sum_{j=i+1}^{M-1} a_{iM}a_{jM}\sigma_{ij}\right)\end{aligned} \qquad \text{[F.5]}$$

Set k equal to the sum of the terms on the second line of equation [F.5],

$$k = \sum_{m=1}^{M-1} a_{mM}\sigma_{mM} + \frac{1}{a_{MM}}\left(\sum_{m=1}^{M-1} a_{mM}^2\sigma_m^2 + 2\sum_{i=1}^{M-2}\sum_{j=i+1}^{M-1} a_{iM}a_{jM}\sigma_{ij}\right), \qquad \text{[F.6]}$$

and set $\hat{k}$ equal to the second term in the enclosed portion of the expression for k,

$$\hat{k} = 2\sum_{i=1}^{M-2}\sum_{j=i+1}^{M-1} a_{iM}a_{jM}\sigma_{ij}.$$

Figure F.1 illustrates the expansion of the expression for $\hat{k}$. Note that the expansion has been written on $M-2$ lines corresponding to the indices of the outer summation term and $M-1$ columns where '$\hat{k}=$' is the first column. Line numbers correspond to the first index on the covariance terms (i.e., the i in σ_{ij}) while column numbers correspond to the second index. Figure F.1 indicates that the expansion for $\hat{k}$ can be equivalently expressed as the

Appendix F Expected Mahalanobis Squared Distances

$$\hat{k} = 2a_{1M}a_{2m}\sigma_{12} + 2a_{1M}a_{3M}\sigma_{13} + \cdots + 2a_{1M}a_{jM}\sigma_{1j} \quad + \cdots + \quad 2a_{1M}a_{(M-2)M}\sigma_{1(M-2)} \quad + \quad 2a_{1M}a_{(M-1)M}\sigma_{1(M-1)}$$

$$+ 2a_{2M}a_{3M}\sigma_{23} + \cdots + 2a_{2M}a_{jM}\sigma_{2j} \quad + \cdots + \quad 2a_{2M}a_{(M-2)M}\sigma_{2(M-2)} \quad + \quad 2a_{2M}a_{(M-1)M}\sigma_{2(M-1)}$$

$$\vdots \qquad\qquad \vdots \qquad\qquad \vdots$$

$$+ 2a_{(j-1)M}a_{jM}\sigma_{(j-1)j} + \cdots + 2a_{(j-1)M}a_{(M-2)M}\sigma_{(j-1)(M-2)} + 2a_{(j-1)M}a_{(M-1)M}\sigma_{(j-1)(M-1)}$$

$$\vdots \qquad\qquad \vdots$$

$$+ 2a_{(M-3)M}a_{(M-2)M}\sigma_{(M-3)(M-2)} + 2a_{(M-3)M}a_{(M-1)M}\sigma_{(M-3)(M-1)}$$

$$+ 2a_{(M-2)M}a_{(M-1)M}\sigma_{(M-2)(M-1)}$$

Figure F.1: Double Summation Expansion

sum of two double summations since each term in the figure is multiplied by two:

$$\hat{k} = \sum_{i=1}^{M-2} \sum_{j=i+1}^{M-1} a_{iM} a_{jM} \sigma_{ij} + \sum_{j=2}^{M-1} \sum_{i=1}^{j-1} a_{iM} a_{jM} \sigma_{ij}. \qquad [F.7]$$

The first double summation in equation [F.7] incorporates one complete set of terms where the inner summation includes all terms in lines $i = 1, \ldots, M-2$ of Figure F.1. The second double summation in equation [F.7] incorporates the remaining complete set of terms where the inner summation includes all terms in columns $j = 2, \ldots, M-1$. Equation [F.7] can be rewritten by swapping indices in the second double summation:

$$\hat{k} = \sum_{i=1}^{M-2} \sum_{j=i+1}^{M-1} a_{iM} a_{jM} \sigma_{ij} + \sum_{i=2}^{M-1} \sum_{j=1}^{i-1} a_{jM} a_{iM} \sigma_{ji}. \qquad [F.8]$$

Then note that the upper limit on the outer summation in the first double summation in equation [F.8] can be increased by one since the inner summation will have no terms when $i = M-1$. Similarly, the lower limit on the outer summation in the second summation in this equation can be decreased by one since the inner summation will have no terms when $i = 1$. Therefore,

$$\hat{k} = \sum_{i=1}^{M-1} \sum_{j=i+1}^{M-1} a_{iM} a_{jM} \sigma_{ij} + \sum_{i=1}^{M-1} \sum_{j=1}^{i-1} a_{iM} a_{jM} \sigma_{ji},$$

which implies that:

$$\hat{k} = \sum_{i=1}^{M-1} a_{iM} \left[\sum_{j=1}^{i-1} a_{jM} \sigma_{ji} + \sum_{j=i+1}^{M-1} a_{jM} \sigma_{ij} \right]. \qquad [F.9]$$

The expression for k in equation [F.6] can then be rewritten as

$$k = \sum_{m=1}^{M-1} \frac{a_{mM}}{a_{MM}} \left[a_{MM} \sigma_{mM} + a_{mM} \sigma_m^2 + \sum_{j=1}^{m-1} a_{jM} \sigma_{jm} + \sum_{j=m+1}^{M-1} a_{jM} \sigma_{mj} \right],$$

or as

$$k = \sum_{m=1}^{M-1} \frac{a_{mM}}{a_{MM}} \left[\sum_{j=1}^{m-1} a_{jM}\sigma_{jm} + a_{mM}\sigma_m^2 + \sum_{j=m+1}^{M-1} a_{jM}\sigma_{mj} + a_{MM}\sigma_{mM} \right], \qquad \text{[F.10]}$$

after rearranging terms. Then, since $\sigma_{mj} = \sigma_{jm}$, the indices on the covariances in the last two terms of equation [F.10] can be reversed yielding:

$$k = \sum_{m=1}^{M-1} \frac{a_{mM}}{a_{MM}} \left[\sum_{j=1}^{m-1} a_{jM}\sigma_{jm} + a_{mM}\sigma_m^2 + \sum_{j=m+1}^{M-1} a_{jM}\sigma_{jm} + a_{MM}\sigma_{Mm} \right], \qquad \text{[F.11]}$$

Now note that the term in brackets in equation [F.11] is zero for all $m = 1, \ldots M-1$ since

$$\sum_{j=1}^{m-1} a_{jM}\sigma_{jm} + a_{mM}\sigma_m^2 + \sum_{j=m+1}^{M-1} a_{jM}\sigma_{jm} + a_{MM}\sigma_{Mm} = \mathbf{a}'_M (\boldsymbol{\Sigma}_M)_{\bullet m} = \left(\boldsymbol{\Sigma}_M^{-1}\right)_{M\bullet} (\boldsymbol{\Sigma}_M)_{\bullet m} = 0.$$

In other words, the bracketed term in equation [F.11] is zero for each $m = 1, \ldots, M-1$ since this term is the inner product of row M from $\boldsymbol{\Sigma}_M^{-1}$ and column m from $\boldsymbol{\Sigma}_M$ with $m \neq M$.

Therefore, $k = 0$ and equation [F.5] can be simplified to:

$$\mathcal{E}\left[\mathcal{M}_{\boldsymbol{\Sigma}_M}(\mathbf{v}, \boldsymbol{\mu})\right] = M - 1 + \left(\sum_{m=1}^{M-1} a_{mM}\sigma_{mM} + a_{MM}\sigma_M^2 \right). \qquad \text{[F.12]}$$

Then note that the enclosed term in equation [F.12] is one since this term is the inner product of row M from $\boldsymbol{\Sigma}_M^{-1}$ and column M from $\boldsymbol{\Sigma}_M$:

$$\sum_{m=1}^{M-1} a_{mM}\sigma_{mM} + a_{MM}\sigma_M^2 = \mathbf{a}'_M (\boldsymbol{\Sigma}_M)_{\bullet M} = \left(\boldsymbol{\Sigma}_M^{-1}\right)_{M\bullet} (\boldsymbol{\Sigma}_M)_{\bullet M} = 1.$$

Hence,

$$\mathcal{E}\left[\mathcal{M}_{\boldsymbol{\Sigma}_M}(\mathbf{v}, \boldsymbol{\mu})\right] = M - 1 + 1 = M,$$

and Proposition 13 has been proven valid by induction. ■

F.3 Proof of Proposition 14

Proposition 14 on page 218 is reproduced below. The proof follows the reproduction of the proposition.

Proposition 14 *Let $\bar{\mathbf{v}} \in \mathbb{R}^M$ be the average random vector for a random sample of size L drawn from a M-variate distribution with mean vector $\boldsymbol{\mu} \in \mathbb{R}^M$ and positive definite covariance matrix $\boldsymbol{\Sigma}_M \in \mathbb{R}^{M \times M}$. Then,*

$$\mathcal{E}\left[\mathcal{M}_{\boldsymbol{\Sigma}_M}\left(\bar{\mathbf{v}}, \boldsymbol{\mu}\right)\right] = \frac{M}{L},$$

when sampling is performed with replacement if the distribution population is discrete.

Proof Let $\mathbf{v}^{(l)}$, $l = 1, \ldots, L$, be the random vectors in the sample such that

$$\bar{\mathbf{v}} = \frac{1}{L} \sum_{l=1}^{L} \mathbf{v}^{(l)}.$$

Proposition 14 is clearly true for $M = 1$ since $\bar{v} = v^{(1)} = v$ implying that

$$\mathcal{E}\left[\mathcal{M}_{\boldsymbol{\Sigma}_1}\left(\bar{v}, \mu\right)\right] = \mathcal{E}\left[\left(v - \mu\right)' \boldsymbol{\Sigma}_1^{-1} \left(v - \mu\right)\right] = \frac{1}{\sigma^2} \mathcal{E}\left[\left(v - \mu\right)^2\right] = \frac{\sigma^2}{\sigma^2} = 1$$

where $\Sigma_1 = [\sigma^2]$. Assume the proposition is true for $M-1$ where $M \geq 2$. Then, equation [F.4] with $\mathbf{v} = \bar{\mathbf{v}}$ and $\mathbf{w} = \boldsymbol{\mu}$ implies that:

$$\begin{aligned}\mathcal{E}\left[\mathcal{M}_{\boldsymbol{\Sigma}_M}\left(\bar{\mathbf{v}}, \boldsymbol{\mu}\right)\right] &= \mathcal{E}\left[\mathcal{M}_{\boldsymbol{\Sigma}_{M-1}}\left(\langle\bar{\mathbf{v}}\rangle_{M-1}, \langle\boldsymbol{\mu}\rangle_{M-1}\right)\right] \\ &+\frac{1}{a_{MM}}\mathcal{E}\left[\sum_{m=1}^{M-1} a_{mM}^2\left(\bar{v}_m - \mu_m\right)^2\right. \\ &\qquad \left.+2\sum_{i=1}^{M-2}\sum_{j=i+1}^{M-1} a_{iM}a_{jM}\left(\bar{v}_i - \mu_i\right)\left(\bar{v}_j - \mu_j\right)\right] \\ &+2\mathcal{E}\left[\sum_{m=1}^{M-1} a_{mM}\left(\bar{v}_m - \mu_m\right)\left(\bar{v}_M - \mu_M\right)\right] + a_{MM}\mathcal{E}\left[\left(\bar{v}_M - \mu_M\right)^2\right].\end{aligned}$$

The induction assumption implies that $\mathcal{E}\left[\mathcal{M}_{\boldsymbol{\Sigma}_{M-1}}\left(\langle\bar{\mathbf{v}}\rangle_{M-1}, \langle\boldsymbol{\mu}\rangle_{M-1}\right)\right] = \frac{M-1}{L}$ so that:

$$\begin{aligned}\mathcal{E}\left[\mathcal{M}_{\boldsymbol{\Sigma}_M}\left(\bar{\mathbf{v}}, \boldsymbol{\mu}\right)\right] &= \frac{M-1}{L} + \frac{1}{a_{MM}}\sum_{m=1}^{M-1} a_{mM}^2\mathcal{E}\left[\left(\bar{v}_m - \mu_m\right)^2\right] \\ &+\frac{2}{a_{MM}}\sum_{i=1}^{M-2}\sum_{j=i+1}^{M-1} a_{iM}a_{jM}\mathcal{E}\left[\left(\bar{v}_i - \mu_i\right)\left(\bar{v}_j - \mu_j\right)\right] \\ &+2\sum_{m=1}^{M-1} a_{mM}\mathcal{E}\left[\left(\bar{v}_m - \mu_m\right)\left(\bar{v}_M - \mu_M\right)\right] + a_{MM}\mathcal{E}\left[\left(\bar{v}_M - \mu_M\right)^2\right],\end{aligned}$$

or,

$$\begin{aligned}\mathcal{E}\left[\mathcal{M}_{\boldsymbol{\Sigma}_M}\left(\bar{\mathbf{v}}, \boldsymbol{\mu}\right)\right] &= \frac{M-1}{L} + \frac{1}{a_{MM}}\sum_{m=1}^{M-1} a_{mM}^2\left[\mathcal{E}\left(\bar{v}_m^2\right) - 2\mu_m\mathcal{E}\left(\bar{v}_m\right) + \mu_m^2\right] \qquad \text{[F.13]} \\ &+\frac{2}{a_{MM}}\sum_{i=1}^{M-2}\sum_{j=i+1}^{M-1} a_{iM}a_{jM}\left[\mathcal{E}\left(\bar{v}_i\bar{v}_j\right) - \mu_j\mathcal{E}\left(\bar{v}_i\right) - \mu_i\mathcal{E}\left(\bar{v}_j\right) + \mu_i\mu_j\right] \\ &+2\sum_{m=1}^{M-1} a_{mM}\left[\mathcal{E}\left(\bar{v}_m\bar{v}_M\right) - \mu_M\mathcal{E}\left(\bar{v}_m\right) - \mu_m\mathcal{E}\left(\bar{v}_M\right) + \mu_m\mu_M\right] \\ &+a_{MM}\left[\mathcal{E}\left(\bar{v}_M^2\right) - 2\mu_M\mathcal{E}\left(\bar{v}_M\right) + \mu_M^2\right].\end{aligned}$$

Since $\mathcal{E}(\bar{v}_m) = \mu_m$ for all $m = 1, \ldots, M$, expectations of the form $\mathcal{E}(\bar{v}_i \bar{v}_j)$ must be evaluated in order to simplify equation [F.13]:

$$\begin{aligned}
\mathcal{E}(\bar{v}_i \bar{v}_j) &= \mathcal{E}\left[\left(\frac{1}{L}\sum_{l_i=1}^{L} v_i^{(l_i)}\right)\left(\frac{1}{L}\sum_{l_j=1}^{L} v_j^{(l_j)}\right)\right] \\
&= \frac{1}{L^2}\mathcal{E}\left(\sum_{l_i=1}^{L}\sum_{l_j=1}^{L} v_i^{(l_i)} v_j^{(l_j)}\right) \\
\mathcal{E}(\bar{v}_i \bar{v}_j) &= \frac{1}{L^2}\sum_{l_i=1}^{L}\sum_{l_j=1}^{L}\mathcal{E}\left(v_i^{(l_i)} v_j^{(l_j)}\right). \qquad \text{[F.14]}
\end{aligned}$$

Expectations in the double summation in equation [F.14] may be evaluated with the well known formula (e.g., see Lindgren [132, Section 4.7, equation (1)]):

$$\mathcal{E}(v_i v_j) = \sigma_{ij} + \mathcal{E}(v_i)\,\mathcal{E}(v_j).$$

Lindgren [132, Section 7.1] states that random vectors $\mathbf{v}^{(l)}$, $l = 1, \ldots, L$, are independent and identically distributed since they are from a random sample. Then, since the covariance between two independent random variables is zero (see Lindgren [132, Section 4.7, Theorem 19]):

$$\mathcal{E}\left(v_i^{(l_i)} v_j^{(l_j)}\right) = \begin{cases} \sigma_{ij} + \mathcal{E}(v_i)\,\mathcal{E}(v_j), & l_i = l_j, \\ \mathcal{E}(v_i)\,\mathcal{E}(v_j), & l_i \neq l_j. \end{cases}$$

Equation [F.14] can now be simplified to:

$$\begin{aligned}
\mathcal{E}(\bar{v}_i \bar{v}_j) &= \frac{1}{L^2}\left[\sum_{l=1}^{L}\left[\sigma_{ij} + \mathcal{E}(v_i)\,\mathcal{E}(v_j)\right] + \sum_{l_i=1}^{L}\sum_{\substack{l_j=1 \\ l_j \neq l_i}}^{L}\mathcal{E}(v_i)\,\mathcal{E}(v_j)\right] \\
&= \frac{1}{L^2}\left[L\sigma_{ij} + L\mu_i\mu_j + L(L-1)\,\mu_i\mu_j\right] \\
\mathcal{E}(\bar{v}_i \bar{v}_j) &= \frac{1}{L}\sigma_{ij} + \mu_i\mu_j. \qquad \text{[F.15]}
\end{aligned}$$

Appendix F Expected Mahalanobis Squared Distances

Equation [F.15] is used to reduce the terms in equation [F.13] that involve expectations as follows:

$$\begin{aligned}\mathcal{E}\left(\bar{v}_m^2\right) - 2\mu_m\mathcal{E}\left(\bar{v}_m\right) + \mu_m^2 &= \frac{1}{L}\sigma_m^2 + \mu_m^2 - 2\mu_m^2 + \mu_m^2 \\ &= \frac{1}{L}\sigma_m^2,\end{aligned}$$

$$\begin{aligned}\mathcal{E}\left(\bar{v}_i\bar{v}_j\right) - \mu_j\mathcal{E}\left(\bar{v}_i\right) - \mu_i\mathcal{E}\left(\bar{v}_j\right) + \mu_i\mu_j &= \frac{1}{L}\sigma_{ij} + \mu_i\mu_j - 2\mu_i\mu_j + \mu_i\mu_j \\ &= \frac{1}{L}\sigma_{ij},\end{aligned}$$

$$\begin{aligned}\mathcal{E}\left(\bar{v}_m\bar{v}_M\right) - \mu_M\mathcal{E}\left(\bar{v}_m\right) - \mu_m\mathcal{E}\left(\bar{v}_M\right) + \mu_m\mu_M &= \frac{1}{L}\sigma_{mM} + \mu_m\mu_M - 2\mu_m\mu_M + \mu_m\mu_M \\ &= \frac{1}{L}\sigma_{mM},\end{aligned}$$

$$\begin{aligned}\mathcal{E}\left(\bar{v}_M^2\right) - 2\mu_M\mathcal{E}\left(\bar{v}_M\right) + \mu_M^2 &= \frac{1}{L}\sigma_M^2 + \mu_M^2 - 2\mu_M^2 + \mu_M^2 \\ &= \frac{1}{L}\sigma_M^2.\end{aligned}$$

Therefore, equation [F.13] can be rewritten as

$$\begin{aligned}\mathcal{E}\left[\mathcal{M}_{\boldsymbol{\Sigma}_M}\left(\bar{\mathbf{v}}, \boldsymbol{\mu}\right)\right] &= \frac{M-1}{L} + \frac{1}{a_{MM}L}\sum_{m=1}^{M-1} a_{mM}^2\sigma_m^2 + \frac{2}{a_{MM}L}\sum_{i=1}^{M-2}\sum_{j=i+1}^{M-1} a_{iM}a_{jM}\sigma_{ij} \\ &\quad + \frac{2}{L}\sum_{m=1}^{M-1} a_{mM}\sigma_{mM} + \frac{1}{L}a_{MM}\sigma_M^2,\end{aligned}$$

or after writing the first term on the second line of the above equation as two summations and regrouping terms,

$$\begin{aligned}\mathcal{E}\left[\mathcal{M}_{\boldsymbol{\Sigma}_M}\left(\bar{\mathbf{v}}, \boldsymbol{\mu}\right)\right] = \frac{1}{L}\Bigg[M - 1 + \left(\sum_{m=1}^{M-1} a_{mM}\sigma_{mM} + a_{MM}\sigma_M^2\right) \\ + \sum_{m=1}^{M-1} a_{mM}\sigma_{mM} + \frac{1}{a_{MM}}\left(\sum_{m=1}^{M-1} a_{mM}^2\sigma_m^2 + 2\sum_{i=1}^{M-2}\sum_{j=i+1}^{M-1} a_{iM}a_{jM}\sigma_{ij}\right)\Bigg].\end{aligned} \qquad [F.16]$$

Now note that the bracketed term in equation [F.16] is the same as the right-hand-side of equation [F.5] on page 404 implying that:

$$\mathcal{E}\left[\mathcal{M}_{\boldsymbol{\Sigma}_M}(\bar{\mathbf{v}}, \boldsymbol{\mu})\right] = \frac{1}{L}\mathcal{E}\left[\mathcal{M}_{\boldsymbol{\Sigma}_M}(\mathbf{v}, \boldsymbol{\mu})\right].$$

Proposition 13 then implies that:

$$\mathcal{E}\left[\mathcal{M}_{\boldsymbol{\Sigma}_M}(\bar{\mathbf{v}}, \boldsymbol{\mu})\right] = \frac{M}{L},$$

and Proposition 14 has been proven valid by induction. ■

F.4 Proof of Proposition 15

Proposition 15 on page 219 is reproduced below. The proof follows the reproduction of the proposition.

Proposition 15 *Let $\bar{\mathbf{v}} \in \mathbb{R}^M$ be the average random vector for a random sample of size L drawn from a M-variate distribution with mean vector $\boldsymbol{\mu} \in \mathbb{R}^M$ and positive definite covariance matrix $\boldsymbol{\Sigma}_M \in \mathbb{R}^{M\times M}$. If $\mathbf{v}$ is a representative vector from the random sample, then*

$$\mathcal{E}\left[\mathcal{M}_{\boldsymbol{\Sigma}_M}(\mathbf{v}, \bar{\mathbf{v}})\right] = \frac{L-1}{L}M,$$

when sampling is performed with replacement if the distribution population is discrete.

Proof Let $\mathbf{v}^{(l)}$, $l = 1, \ldots, L$, be the random vectors in the sample such that

$$\bar{\mathbf{v}} = \frac{1}{L}\sum_{l=1}^{L}\mathbf{v}^{(l)},$$

and let $\mathbf{v}$ be a representative vector from the sample. Proposition 15 is clearly true for $M = 1$ since $\bar{v} = v^{(1)} = v$ implying that

$$\mathcal{E}\left[\mathcal{M}_{\boldsymbol{\Sigma}_1}(v, \bar{v})\right] = \mathcal{E}\left[(v - \bar{v})' \boldsymbol{\Sigma}_1^{-1} (v - \bar{v})\right] = \frac{1}{\sigma^2}\mathcal{E}\left[(v - v)^2\right] = \frac{0}{\sigma^2} = 0$$

where $\boldsymbol{\Sigma}_1 = [\sigma^2]$. Assume the proposition is true for $M - 1$ where $M \geq 2$. Then, equation [F.4] with $\mathbf{w} = \bar{\mathbf{v}}$ implies that:

$$\begin{aligned}
\mathcal{E}\left[\mathcal{M}_{\boldsymbol{\Sigma}_M}(\mathbf{v}, \bar{\mathbf{v}})\right] &= \mathcal{E}\left[\mathcal{M}_{\boldsymbol{\Sigma}_{M-1}}\left(\langle \mathbf{v} \rangle_{M-1}, \langle \bar{\mathbf{v}} \rangle_{M-1}\right)\right] \\
&\quad + \frac{1}{a_{MM}}\mathcal{E}\left[\sum_{m=1}^{M-1} a_{mM}^2 (v_m - \bar{v}_m)^2 \right. \\
&\qquad \left. + 2\sum_{i=1}^{M-2}\sum_{j=i+1}^{M-1} a_{iM} a_{jM} (v_i - \bar{v}_i)(v_j - \bar{v}_j)\right] \\
&\quad + 2\mathcal{E}\left[\sum_{m=1}^{M-1} a_{mM}(v_m - \bar{v}_m)(v_M - \bar{v}_M)\right] + a_{MM}\mathcal{E}\left[(v_M - \bar{v}_M)^2\right].
\end{aligned}$$

The induction assumption implies that $\mathcal{E}\left[\mathcal{M}_{\boldsymbol{\Sigma}_{M-1}}\left(\langle \mathbf{v} \rangle_{M-1}, \langle \bar{\mathbf{v}} \rangle_{M-1}\right)\right] = \frac{L-1}{L}(M-1)$ so that:

$$\begin{aligned}
\mathcal{E}\left[\mathcal{M}_{\boldsymbol{\Sigma}_M}(\mathbf{v}, \bar{\mathbf{v}})\right] &= \frac{L-1}{L}(M-1) + \frac{1}{a_{MM}}\sum_{m=1}^{M-1} a_{mM}^2 \mathcal{E}\left[(v_m - \bar{v}_m)^2\right] \\
&\quad + \frac{2}{a_{MM}}\sum_{i=1}^{M-2}\sum_{j=i+1}^{M-1} a_{iM} a_{jM}\mathcal{E}\left[(v_i - \bar{v}_i)(v_j - \bar{v}_j)\right] \\
&\quad + 2\sum_{m=1}^{M-1} a_{mM}\mathcal{E}\left[(v_m - \bar{v}_m)(v_M - \bar{v}_M)\right] + a_{MM}\mathcal{E}\left[(v_M - \bar{v}_M)^2\right],
\end{aligned}$$

or,

$$\mathcal{E}\left[\mathcal{M}_{\boldsymbol{\Sigma}_M}(\mathbf{v},\bar{\mathbf{v}})\right] = \frac{L-1}{L}(M-1) + \frac{1}{a_{MM}}\sum_{m=1}^{M-1} a_{mM}^2\left[\mathcal{E}\left(v_m^2\right) - 2\mathcal{E}\left(v_m\bar{v}_m\right) + \mathcal{E}\left(\bar{v}_m^2\right)\right] \qquad \text{[F.17]}$$

$$+ \frac{2}{a_{MM}}\sum_{i=1}^{M-2}\sum_{j=i+1}^{M-1} a_{iM}a_{jM}\left[\mathcal{E}\left(v_iv_j\right) - \mathcal{E}\left(\bar{v}_iv_j\right) - \mathcal{E}\left(v_i\bar{v}_j\right) + \mathcal{E}\left(\bar{v}_i\bar{v}_j\right)\right]$$

$$+ 2\sum_{m=1}^{M-1} a_{mM}\left[\mathcal{E}\left(v_mv_M\right) - \mathcal{E}\left(\bar{v}_mv_M\right) - \mathcal{E}\left(v_m\bar{v}_M\right) + \mathcal{E}\left(\bar{v}_m\bar{v}_M\right)\right]$$

$$+ a_{MM}\left[\mathcal{E}\left(v_M^2\right) - 2\mathcal{E}\left(v_M\bar{v}_M\right) + \mathcal{E}\left(\bar{v}_M^2\right)\right].$$

Since equation [F.15] can be used to evaluate expectations of the form $\mathcal{E}\left(\bar{v}_i\bar{v}_j\right)$, evaluations of expectations with the form $\mathcal{E}\left(v_i\bar{v}_j\right)$ are necessary to simplify equation [F.17]:

$$\mathcal{E}\left(v_i\bar{v}_j\right) = \mathcal{E}\left[v_i\left(\frac{1}{L}\sum_{l=1}^{L} v_j^{(l)}\right)\right]$$

$$= \frac{1}{L}\mathcal{E}\left(v_i\sum_{l=1}^{L} v_j^{(l)}\right)$$

$$\mathcal{E}\left(v_i\bar{v}_j\right) = \frac{1}{L}\sum_{l=1}^{L}\mathcal{E}\left(v_iv_j^{(l)}\right). \qquad \text{[F.18]}$$

Then, since $\mathbf{v}$ is a representative vector from the sample, there exists $\hat{l}$ such that $1 \leq \hat{l} \leq L$ and $\mathbf{v} = \mathbf{v}^{(\hat{l})}$. Remarks in the previous section show that vectors $\mathbf{v}^{(l)}$, $l = 1,\ldots,L$, are independent and identically distributed so that:

$$\mathcal{E}\left(v_iv_j^{(l)}\right) = \begin{cases} \sigma_{ij} + \mathcal{E}\left(v_i\right)\mathcal{E}\left(v_j\right), & l = \hat{l}, \\ \mathcal{E}\left(v_i\right)\mathcal{E}\left(v_j\right), & l \neq \hat{l}. \end{cases}$$

Equation [F.18] can now be simplified to:

$$\begin{aligned}\mathcal{E}\left(v_i\bar{v}_j\right) &= \frac{1}{L}\left[\sigma_{ij} + \mathcal{E}\left(v_i\right)\mathcal{E}\left(v_j\right) + \sum_{\substack{l=1\\ l\neq\hat{l}}}^{L}\mathcal{E}\left(v_i\right)\mathcal{E}\left(v_j\right)\right] \\ &= \frac{1}{L}\left[L\sigma_{ij} + \mu_i\mu_j + \left(L-1\right)\mu_i\mu_j\right] \\ \mathcal{E}\left(v_i\bar{v}_j\right) &= \frac{1}{L}\sigma_{ij} + \mu_i\mu_j. \qquad \text{[F.19]}\end{aligned}$$

Equations [F.15] and [F.19] are used to reduce the terms in equation [F.17] that involve expectations as follows:

$$\begin{aligned}\mathcal{E}\left(v_m^2\right) - 2\mathcal{E}\left(v_m\bar{v}_m\right) + \mathcal{E}\left(\bar{v}_m^2\right) &= \sigma_m^2 + \mu_m^2 - 2\left(\frac{1}{L}\sigma_m^2 + \mu_m^2\right) + \frac{1}{L}\sigma_m^2 + \mu_m^2 \\ &= \frac{L-1}{L}\sigma_m^2,\end{aligned}$$

$$\begin{aligned}\mathcal{E}\left(v_iv_j\right) - \mathcal{E}\left(\bar{v}_iv_j\right) - \mathcal{E}\left(v_i\bar{v}_j\right) + \mathcal{E}\left(\bar{v}_i\bar{v}_j\right) &= \sigma_{ij} + \mu_i\mu_j - 2\left(\frac{1}{L}\sigma_{ij} + \mu_i\mu_j\right) + \frac{1}{L}\sigma_{ij} + \mu_i\mu_j \\ &= \frac{L-1}{L}\sigma_{ij},\end{aligned}$$

$$\begin{aligned}\mathcal{E}\left(v_mv_M\right) - \mathcal{E}\left(\bar{v}_mv_M\right) - \mathcal{E}\left(v_m\bar{v}_M\right) + \mathcal{E}\left(\bar{v}_m\bar{v}_M\right) &= \sigma_{mM} + \mu_m\mu_M - 2\left(\frac{1}{L}\sigma_{mM} + \mu_m\mu_M\right) \\ &\quad + \frac{1}{L}\sigma_{mM} + \mu_m\mu_M \\ &= \frac{L-1}{L}\sigma_{mM},\end{aligned}$$

$$\begin{aligned}\mathcal{E}\left(v_M^2\right) - 2\mathcal{E}\left(v_M\bar{v}_M\right) + \mathcal{E}\left(\bar{v}_M^2\right) &= \sigma_M^2 + \mu_M^2 - 2\left(\frac{1}{L}\sigma_M^2 + \mu_M^2\right) + \frac{1}{L}\sigma_M^2 + \mu_M^2 \\ &= \frac{L-1}{L}\sigma_M^2.\end{aligned}$$

Therefore, equation [F.17] can be rewritten as

$$\begin{aligned}\mathcal{E}\left[\mathcal{M}_{\mathbf{\Sigma}_M}(\mathbf{v},\bar{\mathbf{v}})\right] &= \frac{L-1}{L}(M-1)+\frac{1}{a_{MM}}\sum_{m=1}^{M-1}a_{mM}^2\left[\mathcal{E}\left(v_m^2\right)-2\mathcal{E}\left(v_m\bar{v}_m\right)+\mathcal{E}\left(\bar{v}_m^2\right)\right]\\ &\quad+\frac{2}{a_{MM}}\sum_{i=1}^{M-2}\sum_{j=i+1}^{M-1}a_{iM}a_{jM}\left[\mathcal{E}\left(v_iv_j\right)-\mathcal{E}\left(\bar{v}_iv_j\right)-\mathcal{E}\left(v_i\bar{v}_j\right)+\mathcal{E}\left(\bar{v}_i\bar{v}_j\right)\right]\\ &\quad+2\sum_{m=1}^{M-1}a_{mM}\left[\mathcal{E}\left(v_mv_M\right)-\mathcal{E}\left(\bar{v}_mv_M\right)-\mathcal{E}\left(v_m\bar{v}_M\right)+\mathcal{E}\left(\bar{v}_m\bar{v}_M\right)\right]\\ &\quad+a_{MM}\left[\mathcal{E}\left(v_M^2\right)-2\mathcal{E}\left(v_M\bar{v}_M\right)+\mathcal{E}\left(\bar{v}_M^2\right)\right].\end{aligned}$$

$$\begin{aligned}\mathcal{E}\left[\mathcal{M}_{\mathbf{\Sigma}_M}(\mathbf{v},\bar{\mathbf{v}})\right] &= \frac{L-1}{L}(M-1)+\frac{L-1}{a_{MM}L}\sum_{m=1}^{M-1}a_{mM}^2\sigma_m^2+\frac{2(L-1)}{a_{MM}L}\sum_{i=1}^{M-2}\sum_{j=i+1}^{M-1}a_{iM}a_{jM}\sigma_{ij}\\ &\quad+\frac{2(L-1)}{L}\sum_{m=1}^{M-1}a_{mM}\sigma_{mM}+\frac{L-1}{L}a_{MM}\sigma_M^2,\end{aligned}$$

or after writing the first term on the second line of the above equation as two summations and regrouping terms,

$$\begin{aligned}\mathcal{E}\left[\mathcal{M}_{\mathbf{\Sigma}_M}(\mathbf{v},\bar{\mathbf{v}})\right] = \frac{L-1}{L}\Bigg[M-1+\left(\sum_{m=1}^{M-1}a_{mM}\sigma_{mM}+a_{MM}\sigma_M^2\right)\\ +\sum_{m=1}^{M-1}a_{mM}\sigma_{mM}+\frac{1}{a_{MM}}\left(\sum_{m=1}^{M-1}a_{mM}^2\sigma_m^2+2\sum_{i=1}^{M-2}\sum_{j=i+1}^{M-1}a_{iM}a_{jM}\sigma_{ij}\right)\Bigg].\end{aligned} \qquad [F.20]$$

Now note that the bracketed term in equation [F.20] is the same as the right-hand-side of equation [F.5] on page 404 implying that:

$$\mathcal{E}\left[\mathcal{M}_{\mathbf{\Sigma}_M}(\mathbf{v},\bar{\mathbf{v}})\right] = \frac{L-1}{L}\mathcal{E}\left[\mathcal{M}_{\mathbf{\Sigma}_M}(\mathbf{v},\boldsymbol{\mu})\right].$$

Proposition 13 then implies that:

$$\mathcal{E}\left[\mathcal{M}_{\mathbf{\Sigma}_M}(\mathbf{v},\bar{\mathbf{v}})\right] = \frac{L-1}{L}M,$$

and Proposition 15 has been proven valid by induction. ■

Appendix G
Two-Stage Decomposition Graphics

This appendix contains graphs illustrating the CPU times required by two-stage decomposition procedures to achieve selected relative tolerance values. Each figure corresponds to one of the multiple period problems described in Section 6.1.2 and listed in Table 6.3 on page 231. Note that problem **P5-Large** is not represented by a figure since this problem could not be solved with two-stage decomposition methods.

Each figure has a curve representing required CPU time versus relative tolerance for each decomposition method that was applied to the applicable problem. Time units are shown as part of the labels for the vertical axes. Relative tolerance values are shown decreasing from left to right in a logarithmic scale along the horizontal axes. The figures are listed below in their order of appearance.

Figure G.1: DWD(7.1.2) and LSD(7.1.2) applied to problem **P2-Small**.

Figure G.2: DWD(7.1.2) and LSD(7.1.2) applied to problem **P2-Medium**.

Figure G.3: DWD(7.1.4) and LSD(7.1.4) applied to problem **P2-Large**.

Figure G.4: DWD(7.1.2), DWD(7.2.2), LSD(7.1.2), and LSD(7.2.2) applied to problem **P3-Small**.

Figure G.5: DWD(7.1.2), DWD(7.2.2), LSD(7.1.2), and LSD(7.2.2) applied to problem **P3-Medium**.

Figure G.6: DWD(7.2.2) and LSD(7.2.2) applied to problem **P3-Large**.

Figure G.7: DWD(7.1.2), DWD(7.2.2), DWD(7.3.2), LSD(7.1.2), LSD(7.2.2), and LSD(7.3.2) applied to problem **P4-Small**.

Figure G.8: DWD(7.2.2), DWD(7.3.2), LSD(7.2.2), and LSD(7.3.2) applied to problem **P4-Medium**.

Figure G.9: DWD(7.2.2) and LSD(7.2.2) applied to problem **P4-Large**.

Figure G.10: DWD(7.2.2), DWD(7.3.2), DWD(7.4.2), LSD(7.2.2), LSD(7.3.2), and LSD(7.4.2) applied to problem **P5-Small**.

Figure G.11: DWD(7.3.2) and LSD(7.3.2) applied to problem **P5-Large**.

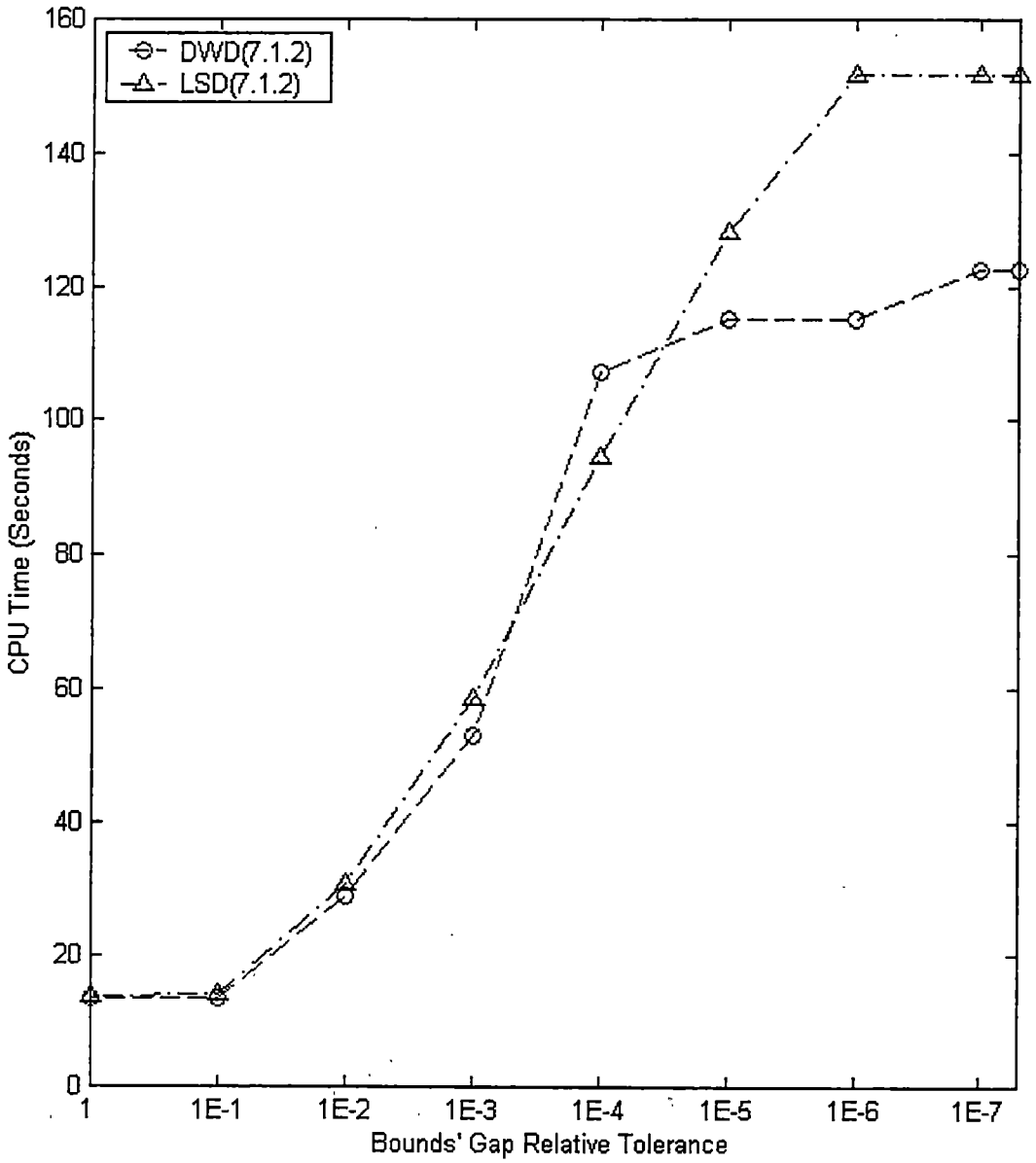


Figure G.1: Two-Stage Decomposition Times for Problem **P2-Small**

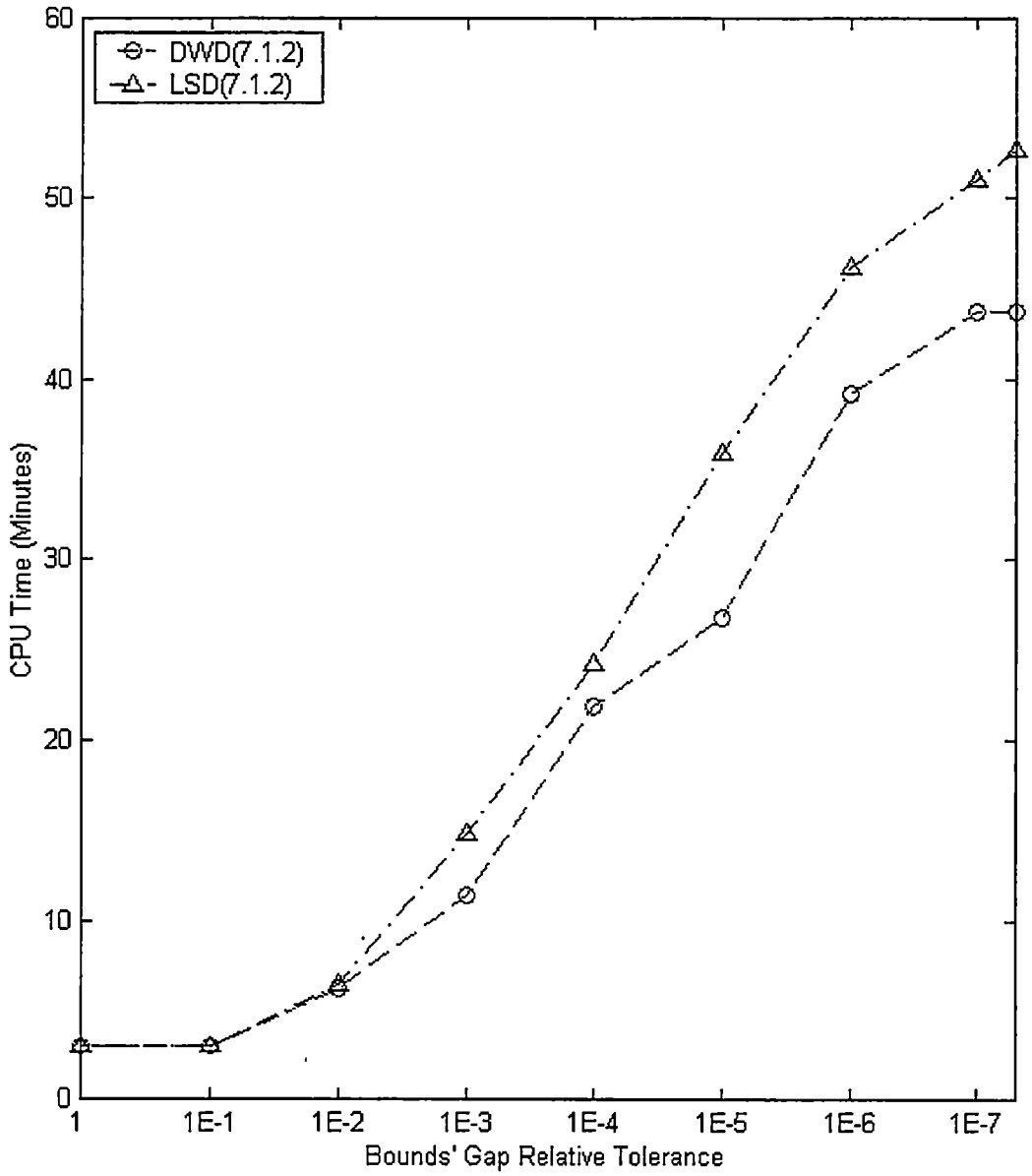


Figure G.2: Two-Stage Decomposition Times for Problem **P2-Medium**

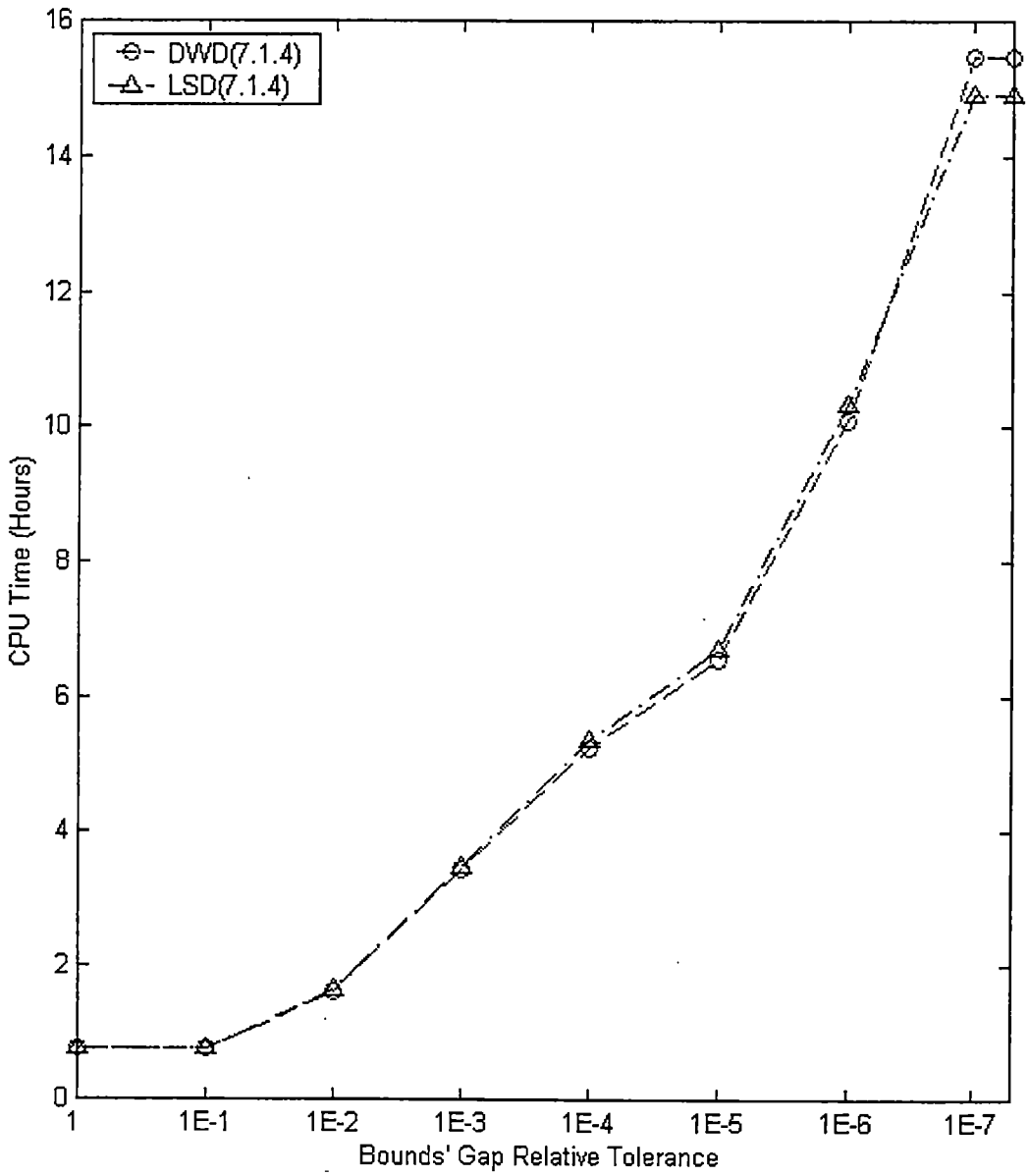


Figure G.3: Two-Stage Decomposition Times for Problem **P2-Large**

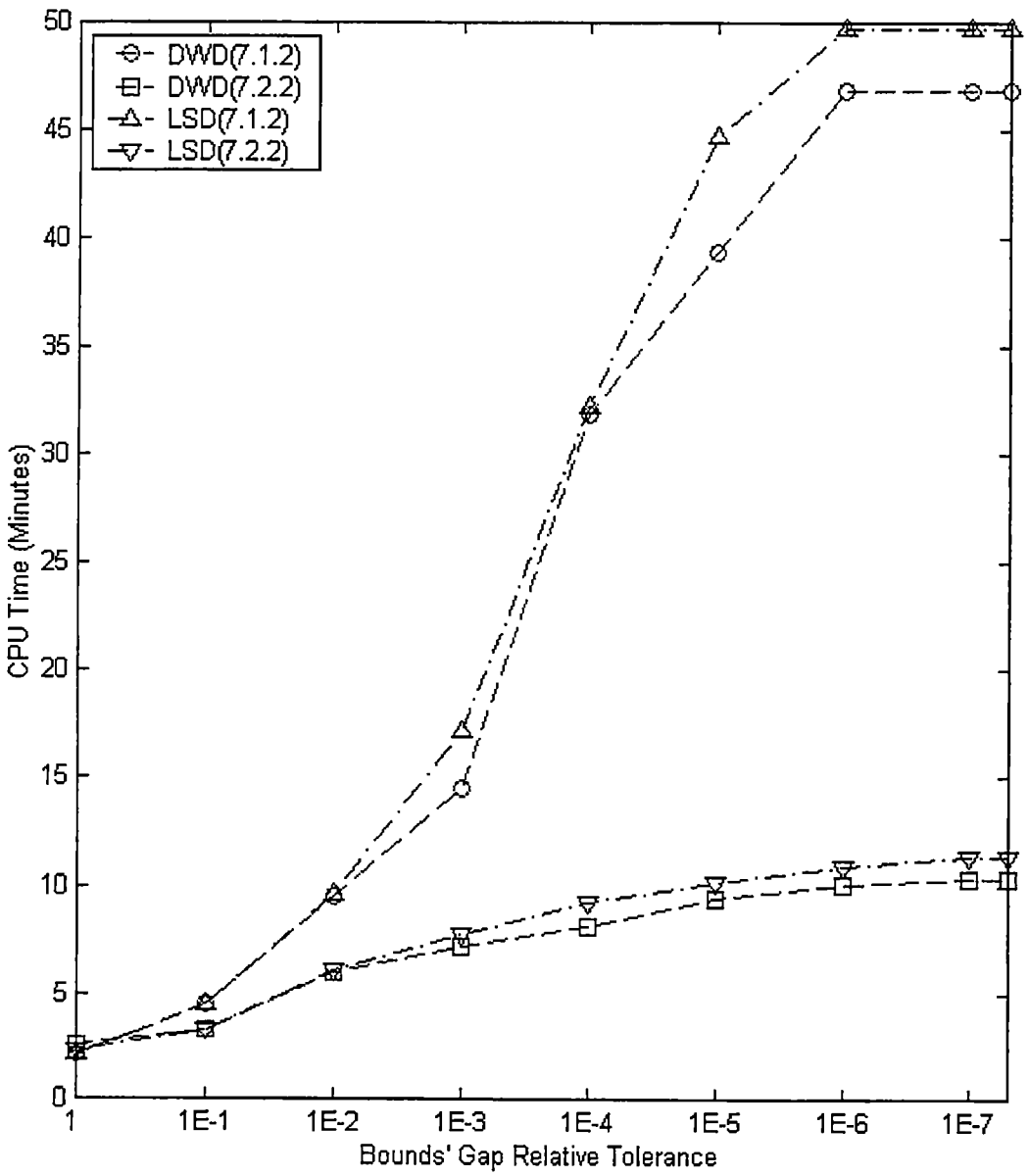


Figure G.4: Two-Stage Decomposition Times for Problem **P3-Small**

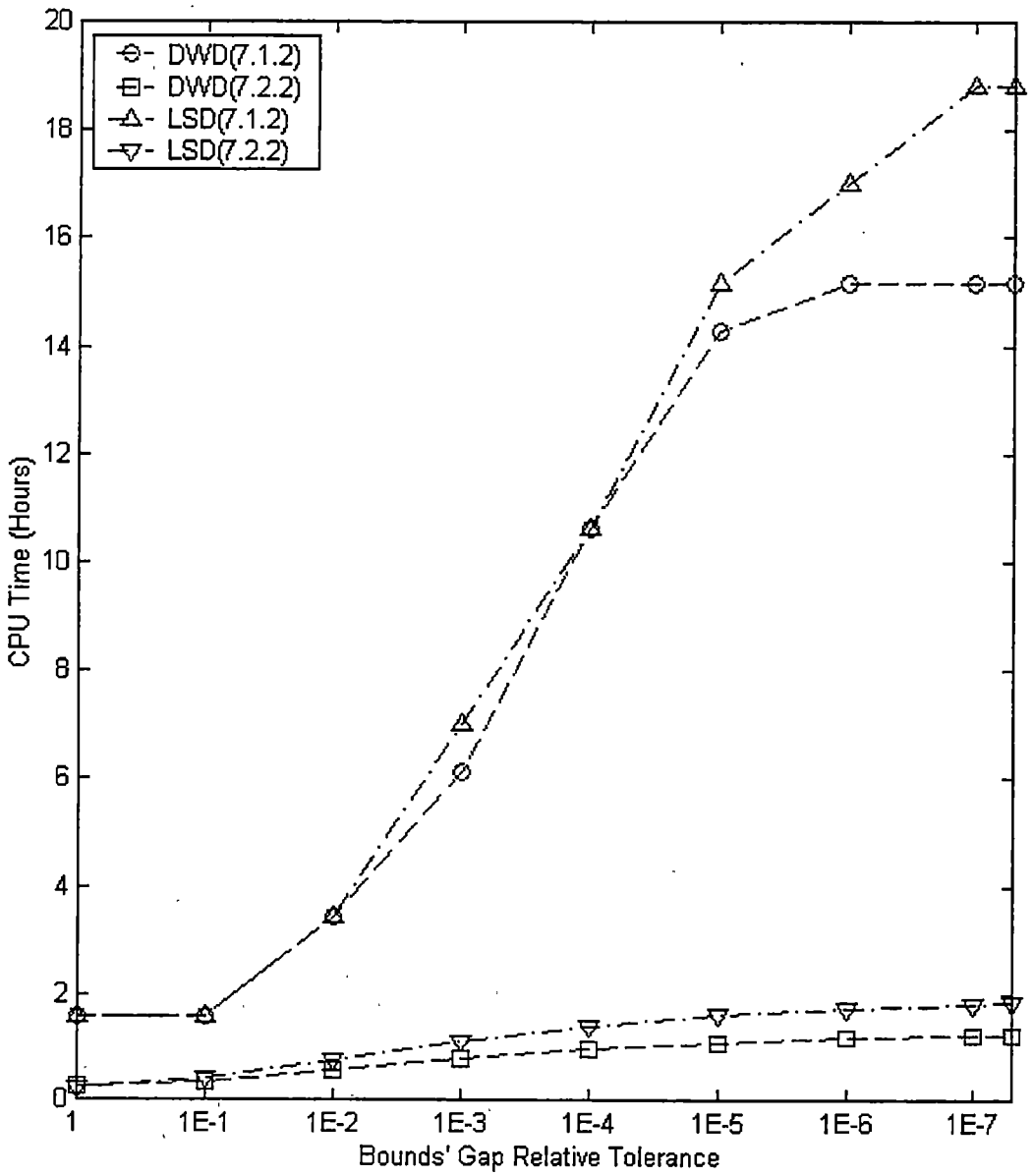


Figure G.5: Two-Stage Decomposition Times for Problem **P3-Medium**

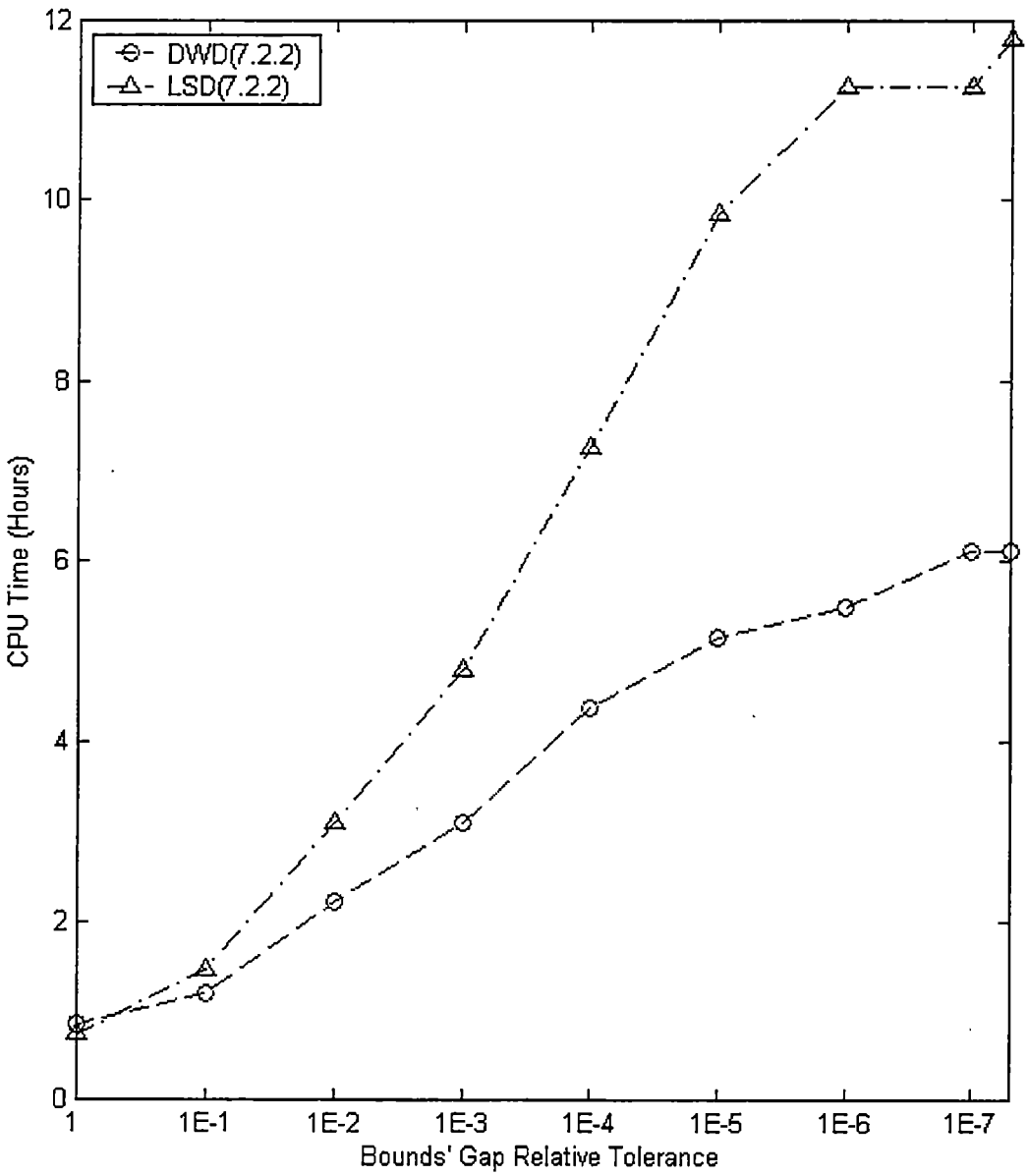


Figure G.6: Two-Stage Decomposition Times for Problem **P3-Large**

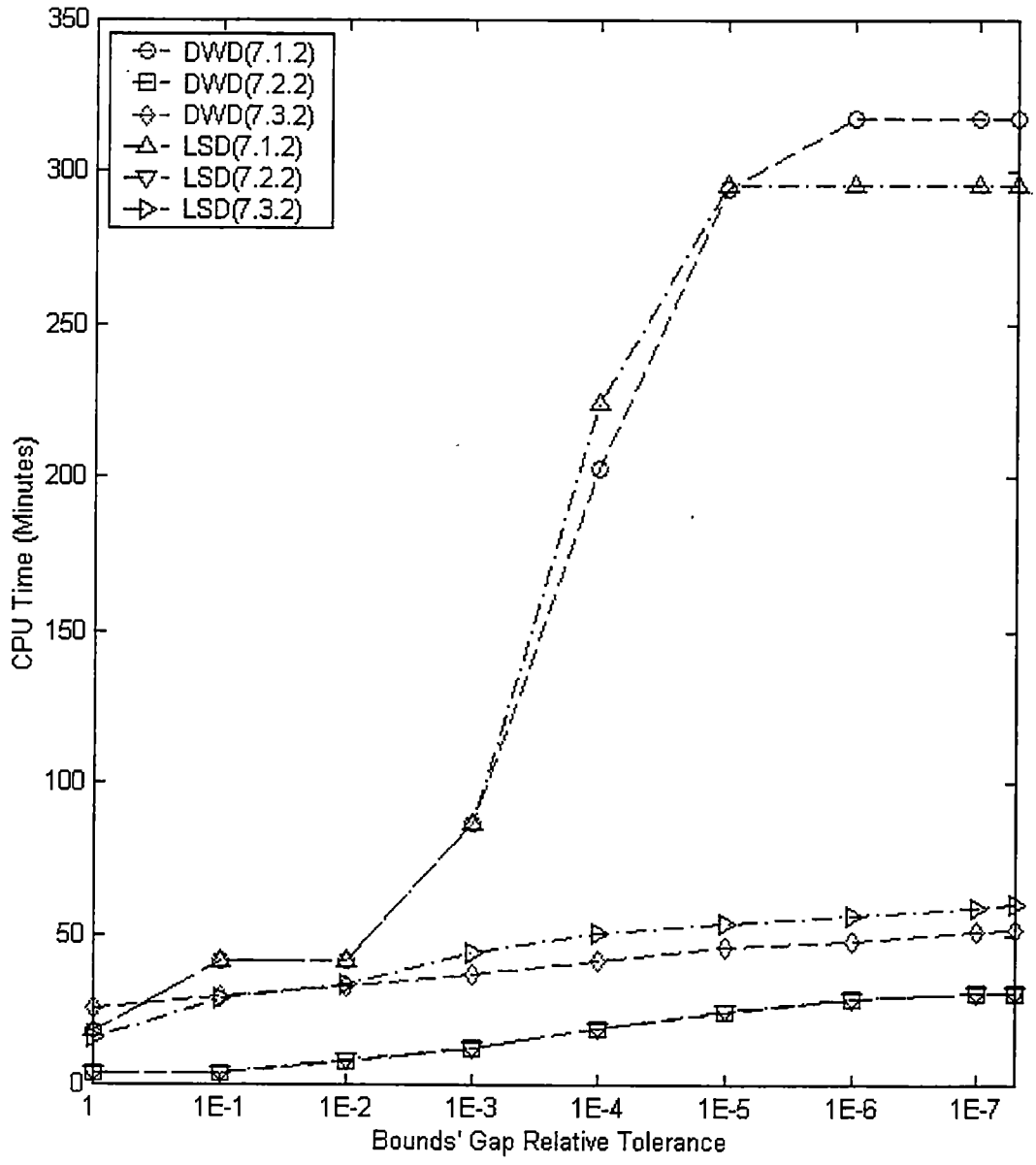


Figure G.7: Two-Stage Decomposition Times for Problem **P4-Small**

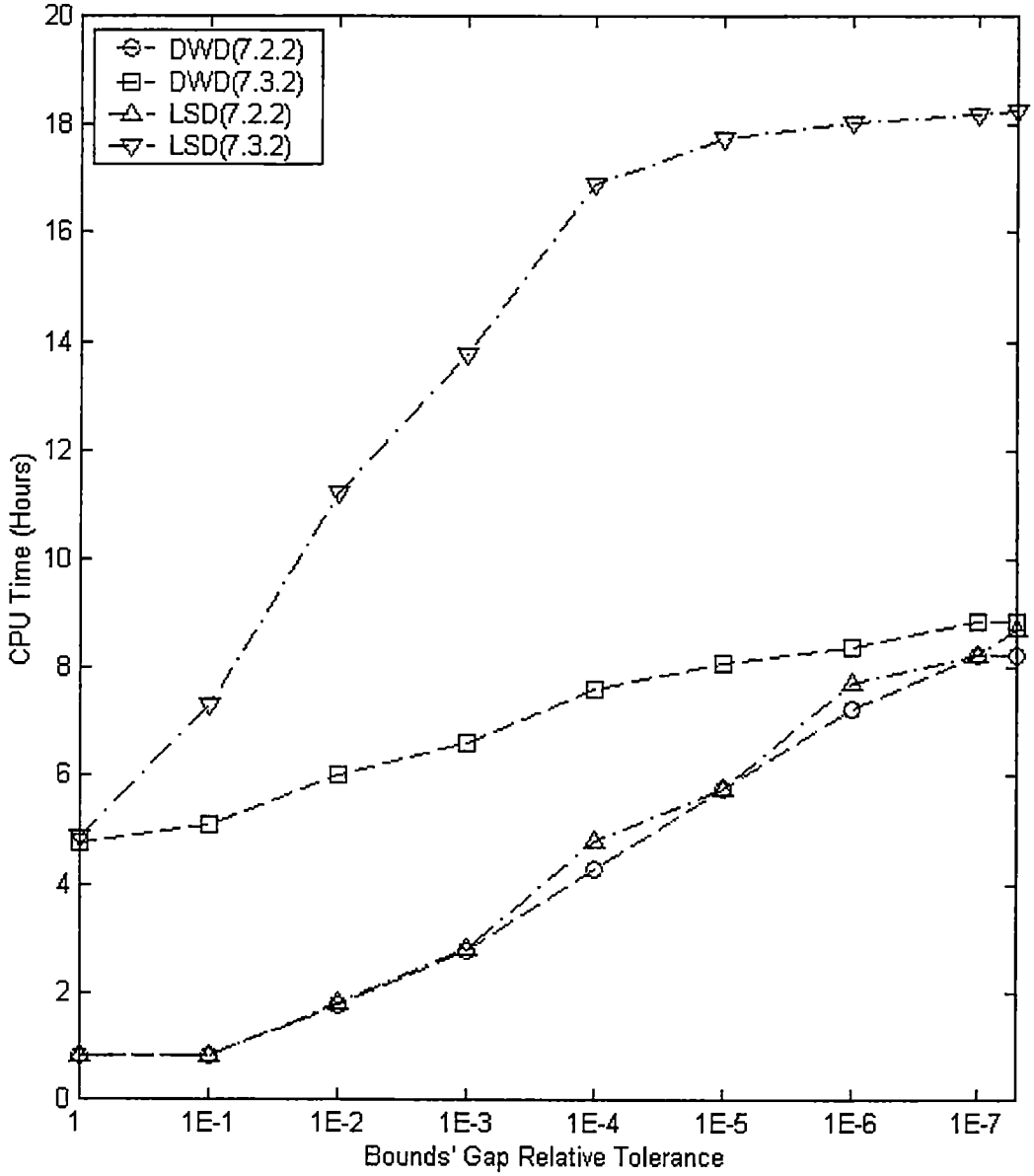


Figure G.8: Two-Stage Decomposition Times for Problem **P4-Medium**

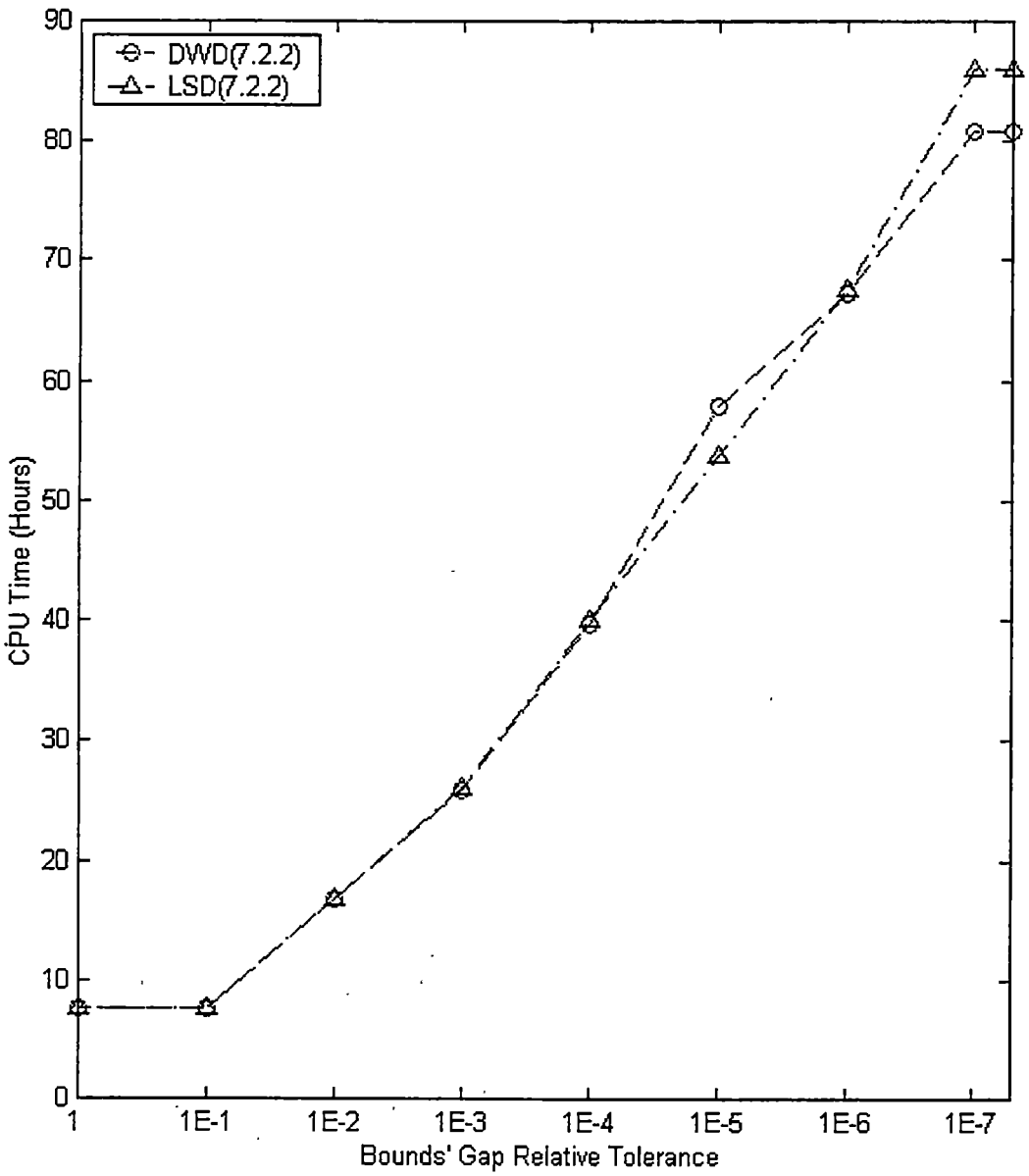


Figure G.9: Two-Stage Decomposition Times for Problem **P4-Large**

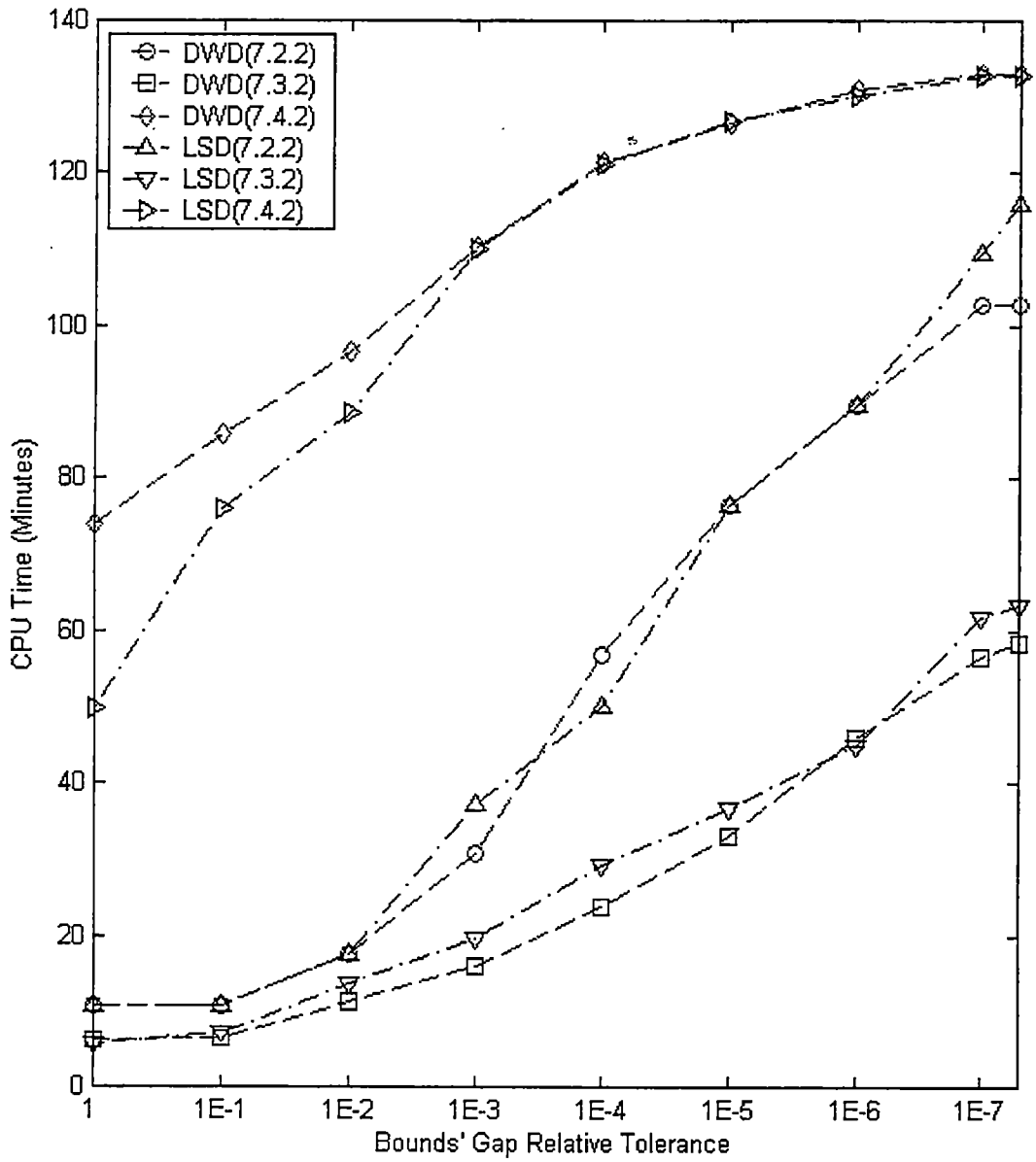


Figure G.10: Two-Stage Decomposition Times for Problem **P5-Small**

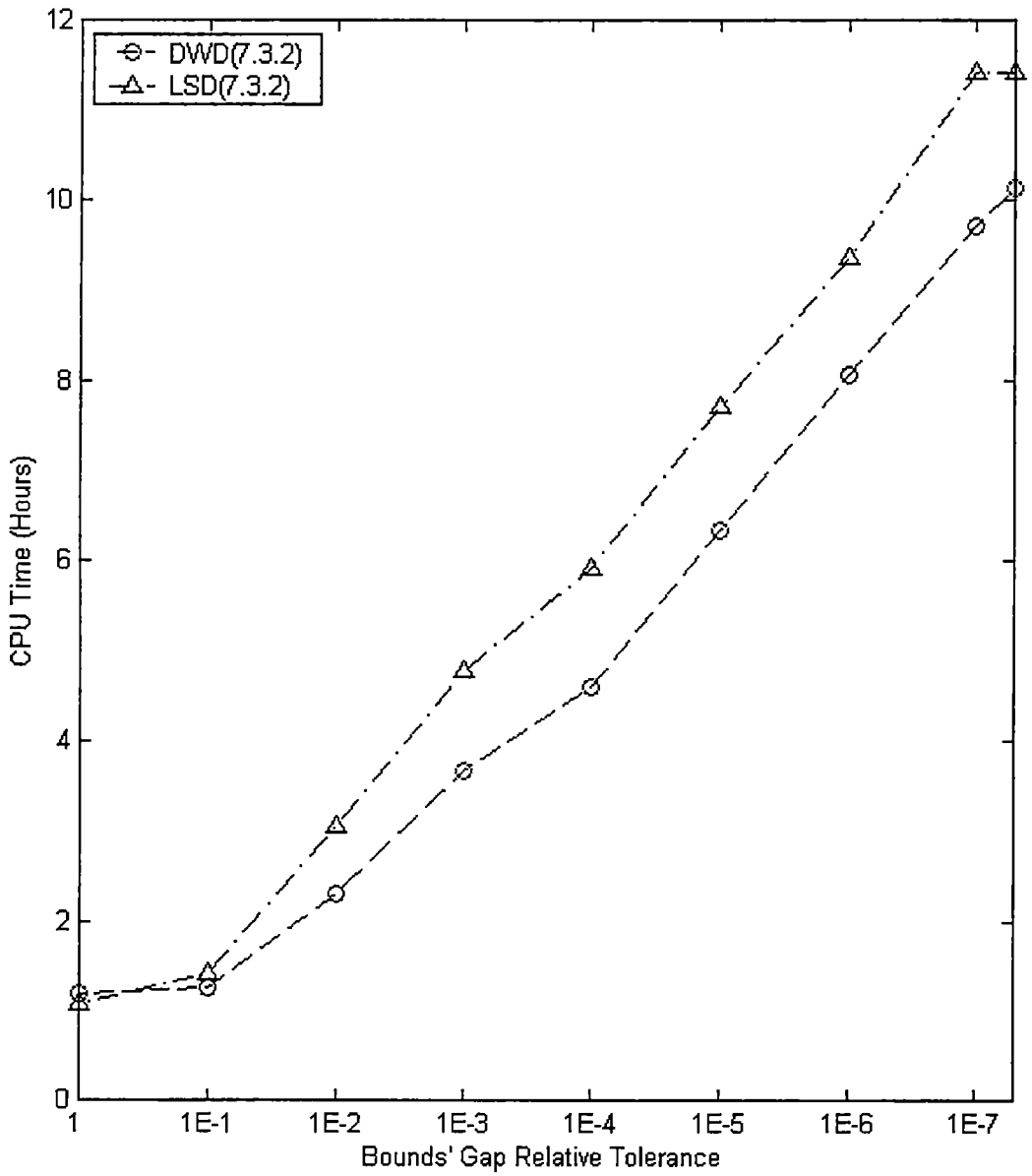


Figure G.11: Two-Stage Decomposition Times for Problem **P5-Medium**

Appendix H
Acronyms

Table H.1: Acronyms

Acronym	Description	Chapter[a]	Page[b]
BF	backwards first	4	145
CPU	central processing unit	4	152
CVF	Compaq Visual Fortran	5	205
DEP	deterministic equivalent problem	2	20
DWD	Dantzig-Wolfe decomposition	1	3
FF	fast forward	4	145
FFFB	fast forward-fast back	4	144
GLP	grand linear program	2	22
GUI	graphical user interface	5	208
LP	linear program	2	21
LSD	L-Shaped decomposition	1	4
MDPCA	myopic dual-primal cycling algorithm	4	153
MIMPSLP[c]	market investment multiple period stocastic linear program	5	164 and 206
MPS	mathematical programming system	1	6
MSD	Mahalanobis squared distance	5	216
PWL	piece-wise linear	5	168
RAM	random access memory	4	157
RMP[d]	relaxed/restricted master problem	3	47 and 77
RMP-SUB[e]	relaxed/restricted master problem-subproblem	4	119
SLP	stochastic linear program with recource	2	22
SUB	subproblem	3	47

[a] Chapter number where first defined

[b] Page number where first defined

[c] **Bold** text for the model or *italic* text for the code library

[d] First letter represents *relaxed* for LSD or *restricted* for DWD

[e] Dual purpose problem at a node in the intermediate stage of a nested decomposition procedure

Vita

Earl I. (Ike) Patterson was born in Maryville, Tennessee on 9 June 1950. Put some biographical info in this section. He attended public schools in Maryville and was admitted to the United States Military Academy (USMA) in July 1968 after graduation from Everett High School. Mr. Patterson graduated from USMA in June 1972 and was commissioned in the Corps of Engineers. Subsequent assignments included platoon leader, company commander, associate professor of mathematics at USMA, and staff positions with the Joint Strategic Target Planning Staff and Army Nuclear and Chemical Agency. He received a Master's degree in Nuclear Engineering from the Massachusetts Institute of Technology in 1982 preparatory to the teaching assignment at USMA. Mr. Patterson retired from the Army in 1992 and was admitted to the Management Science doctoral program at the University of Tennessee in the fall of 1994. His doctoral work was completed in May 2001 and he currently resides in his home town of Maryville.