INTERNATIONAL SERIES ON
APPLIED SYSTEMS ANALYSIS

SYSTEMS ANALYSIS BY MULTILEVEL METHODS: With Applications to Economics and Management

YVO M.I. DIRICKX
L. PETER JENNERGREN

International Institute for
Applied Systems Analysis

SYSTEMS ANALYSIS BY MULTILEVEL METHODS:
With Applications to Economics and Management

Yvo M.I. Dirickx
Department of Applied Mathematics
Twente University of Technology

and

L. Peter Jennergren
Department of Business Administration
Odense University

The book presents a survey of usable multilevel methods for modeling and solving decision problems in economics and management. The methods are largely extensions of linear programming and fall within the realm of column generation and decomposition. About one third of the book is concerned with methods and the rest describes case studies where these methods have actually been used. They are taken from areas such as national and regional economic planning, production planning, and transportation planning.

Contents

1 Introduction

2 Fundamental concepts

3 Multilevel solution methods

4 Numerical experiences with Dantzig–Wolfe decomposition

5 National and regional economic planning

6 Planning of production and sales programs in corporations

7 Operations management

8 Distribution systems

9 Freight ship route scheduling and electricity generation

10 Water pollution control

11 Conclusion

6 International Series on
Applied Systems Analysis

Systems Analysis by Multilevel Methods

With Applications to Economics and Management

Yvo M. I. Dirickx
Department of Applied Mathematics,
Twente University of Technology

L. Peter Jennergren
Department of Business Administration,
Odense University

A Wiley–Interscience Publication
International Institute for Applied Systems Analysis

JOHN WILEY & SONS

Chichester—New York—Brisbane—Toronto

British Library Cataloguing in Publication Data

Dirickx, Yvo M I
Systems analysis by multilevel methods.
—(International series on applied systems analysis).
1. Management 2. System analysis
3. Economics, Mathematical
I. Title II. Jennergren, L Peter
III. Series
658.4'032 HD38 79-40639

ISBN 0 471 27626 X

Printed in England by The Pitman Press, Bath

Preface

In this volume, we present a survey of multilevel systems analysis methods and selected applications of these methods in economics and management. The first part of the book (Chapters 1–4) offers some theories and methods of multilevel systems analysis, and later chapters (5–10) deal with concrete applications to such areas as national and regional planning and industrial management. In line with the objectives of the International Institute for Applied Systems Analysis (IIASA) Survey Project, this volume develops no new theory, but outlines concepts and methods that have proven useful in applications. The sample of application areas presented is rather broad and, we hope, representative of the existing literature dealing with multilevel systems analysis in economics and management.

So as not to mislead the reader, we point out here that more engineering-oriented applications of multilevel systems analysis (e.g., to chemical process control) are not discussed in this volume. As a consequence of our focus on applications in economics and management, the methods outlined in the book fall within the realm of decomposition in mathematical programming. In fact, much of the book is based on linear programming and immediate extensions thereof. It follows then that the mathematical prerequisites for an understanding of the book are modest: a knowledge of linear programming and some familiarity with nonlinear and dynamic programming.

The intended audience for this volume consists of operations researchers and systems analysts in government and industry. We also hope that it will be useful to students in fields such as operations research, economics, and management. On the basis of teaching experiences with a preliminary version of the book at the Catholic University of Louvain, Bielefeld University, and Odense University, we believe it can be used for a one-semester course in multilevel systems analysis, if supplemented with exercises and additional readings.

During the process of writing this volume, we have been associated with the Catholic University of Louvain, the European Institute for Advanced Studies in Management (Brussels), and Bielefeld University (Y. D.); Odense University, and IIASA (P. J.). We wish to thank our colleagues and students at these institutions for valuable suggestions. In particular, we wish to thank the following individuals for suggestions and comments: Willy Gochet and Marc Lambrecht of the Catholic University of Louvain; Børge Obel of Odense University; and Janusz Kindler, William Orchard-Hays, Jan W. Owsinki, and Andrej Straszak of IIASA. We also wish to thank Søren Glud Johansen of Århus University, Oli B. G. Madsen of the Technical University of Denmark, Adam Wozniak of the Technical University of Warsaw, and the two outside reviewers of our manuscript. Furthermore, we extend thanks to the following members of the IIASA Survey Project: Giandomenico Majone, Edward S. Quade, and Vil Z. Rakhmankulov. We are also grateful to Wladyslaw Findeisen, our scientific editor, who has followed our work from the beginning, making very valuable suggestions all along the way, and to Jeannette Lindsay, our technical editor. Finally, we thank Annie Gertz of Odense University for a fine typing job.

YVO M. I. DIRICKX
L. PETER JENNERGREN

Contents

1 Introduction 1
- 1.1 A first look at multilevel systems analysis 1
- 1.2 The multilevel character of systems 3
- 1.3 The historical development of multilevel systems analysis 4
- 1.4 Overview of the volume 6
- References 9

2 Fundamental Concepts 10
- 2.1 The idealized multilevel approach 10
  - *2.1.1 Partitioning the overall problem into subproblems,* 10
  - *2.1.2 Examples of two-level subproblem hierarchies,* 12
  - *2.1.3 The multilevel solution process,* 17
- 2.2 Additional aspects of multilevel systems analysis 19
  - *2.2.1 A more general multilevel approach,* 19
  - *2.2.2 Institutional interpretations and analogies,* 22
  - *2.2.3 Related concepts,* 23
- References 25

3 Multilevel Solution Methods 27
- 3.1 Introduction and overview 27
- 3.2 Column generation 28
  - *3.2.1 General discussion,* 28
  - *3.2.2 The maximal multicommodity network flow problem,* 31
  - *3.2.3 Solution by column generation,* 33
- 3.3 The Dantzig–Wolfe decomposition method for linear programs 36
  - *3.3.1 The representation of a polyhedral convex set,* 37

3.3.2 An outline of the Dantzig–Wolfe decomposition method, 37
3.3.3 Linear programming problems with unbounded solutions, 41
3.3.4 A numerical example, 43
3.3.5 Some further remarks on the Dantzig–Wolfe decomposition method, 45
3.3.6 Block-angular structures, 48
3.4 The Dantzig–Wolfe method for nonlinear programs 54
3.5 The Benders algorithm and some extensions 56
3.5.1 An outline of the Benders algorithm, 56
3.5.2 A note on Step 2 of the Benders algorithm, 61
3.5.3 A numerical example, 63
3.5.4 The application of the Benders algorithm to block-angular structures, 66
3.5.5 On the relation between the Benders and Dantzig–Wolfe algorithms, 69
3.6 The Kornai–Liptak decomposition algorithm 70
3.7 Lagrangean decomposition in nonlinear programming 72
3.7.1 Lagrangean decomposition for separable mathematical programming problems, 73
3.7.2 Duality theory and Lagrangean decomposition, 75
3.8 Heuristic methods 76
3.9 Multilevel control theory: a brief survey 76
3.9.1 Static multilevel control problems, 77
3.9.2 Dynamic open-loop multilevel control, 78
3.9.3 On-line control models, 81
References 81

4 Numerical Experiences with Dantzig–Wolfe decomposition 84
4.1 On the utilization of structure in solving linear programming problems 84
4.2 Test problem experiences 86
References 97

5 National and Regional Economic Planning 99
5.1 Introduction and overview 99
5.2 Multilevel national economic planning in Hungary 101
5.2.1 The application of the Kornai–Liptak method to a national economic planning problem, 101
5.2.2 The application of man–machine planning to the 1966–1970 5-year plan, 105
5.2.3 Concluding remarks, 111

5.3 Multilevel national economic planning in Mexico 112
5.3.1 Introduction, 112
5.3.2 DINAMICO, 115
5.3.3 ENERGETICOS, 117
5.3.4 Linkages between DINAMICO *and* ENERGETICOS, 120
5.3.5 Conclusions and comparison with multilevel national economic planning in Hungary, 123
5.4 A problem of regional planning 125
5.4.1 The development network, 125
5.4.2 An LP *model for resource production*, 127
5.4.3 The overall problem and a two-level solution method, 128
5.4.4 Discussion of the two-level method for regional planning, 129
References 130

6 Planning of Production and Sales Programs in Corporations 132
6.1 Introduction 132
6.1.1 The planning problem, 132
6.1.2 Planning procedures based on decomposition methods, 133
6.2 A simulation study of a planning procedure based on the Dantzig–Wolfe method in a paperboard factory 137
6.2.1 The planning problem of the paperboard factory, 137
6.2.2 Information dispersal and information flows, 140
6.2.3 The simulation experiment, 141
6.2.4 Implementation of the plan, 142
6.2.5 Some conclusions, 144
6.3 A simulation study of planning procedures based on the Dantzig–Wolfe and ten Kate methods in a slaughterhouse 145
6.3.1 The planning problem of the slaughterhouse, 145
6.3.2 Simulated results using the Dantzig–Wolfe method as a planning procedure, 147
6.3.3 Simulated results using the ten Kate method as a planning procedure, 149
6.3.4 Some conclusions, 152
6.4 Final remarks on planning procedures based on decomposition methods 153
References 154

7 Operations Management 156
7.1 Introduction and overview 156
7.2 A column generation approach 158
7.2.1 An approximating LP *problem*, 158

7.2.2 Generation of dominant schedules and a two-level algorithm, 159
7.2.3 Applications, 161
7.3 Hierarchical production planning 162
7.3.1 Introduction to hierarchical production planning, 162
7.3.2 A three-level disaggregation scheme, 162
7.3.3 The product-type-level subproblem, 163
7.3.4 The item-family-level subproblems, 164
7.3.5 The item-level subproblems, 168
7.3.6 A three-level solution procedure, 168
7.3.7 Applications and a comparison with column generation, 170
References 171

8 Distribution Systems 172
8.1 Introduction and overview 172
8.2 The optimal design of a distribution system 173
8.2.1 A mixed-integer programming formulation, 173
8.2.2 Application of the Benders algorithm, 174
8.2.3 The implementation of Geoffrion and Graves, 176
8.3 Determining optimal production–distribution programs 179
8.3.1 A network flow formulation, 179
8.3.2 A column generation method, 180
8.3.3 The implementation of Folie and Tiffin, 181
References 181

9. Freight Ship Route Scheduling and Electricity Generation 183
9.1 Introduction and overview 183
9.2 Freight ship route scheduling 183
9.2.1 Problem formulation, 183
9.2.2 The generation of ship itineries, 185
9.2.3 A column generation scheme, 187
9.2.4 A three-level method, 189
9.3 Planning power generation 190
9.3.1 Problem formulation, 190
9.3.2 Application of the Dantzig–Wolfe method, 191
References 193

10 Water Pollution Control 194
10.1 Introduction and overview 194
10.2 The Miami River case 195
10.2.1 The overall problem, 195

10.2.2 A planning procedure based on Dantzig–Wolfe decomposition, 197
10.2.3 A Lagrangean solution method, 200
10.3 Concluding remarks 202
References 203

11 Conclusion 205
11.1 Problem structures and solution methods 205
11.2 An evaluation of the usefulness of multilevel methods 208
11.3 A final word 213
References 213

Index 215

1 Introduction

1.1 A FIRST LOOK AT MULTILEVEL SYSTEMS ANALYSIS

Multilevel systems analysis is a somewhat obscure designation for a methodology that has been added to the systems analyst's bag of tricks over the last ten or fifteen years. In this volume, we will take a closer look at that methodology. By way of introduction, this chapter describes informally what we mean by multilevel systems analysis. It also comments on the fact that many systems are hierarchical (in particular, organizations) traces the historical development of multilevel systems analysis, and outlines the contents of the volume.

We are here concerned with decision problems arising in economics and management. As examples of the type of decision problems that we have in mind, and which are discussed in later chapters, one may cite the following:

Decide on a comprehensive development plan for a newly opened economic region, specifying, for instance, which investment projects are to be undertaken.

Determine a production schedule for a factory, stating what quantity of each product is to be produced over, say, the next month.

Compose a route plan for a shipping line, specifying for each ship a cargo and a destination.

When faced with a decision problem such as these, the analyst usually starts out by constructing an idealized representation, a model, of the problem situation, often in the form of an optimization formulation. This model can then be manipulated in various ways. In the course of such manipulations, a solution to the decision problem under consideration may be obtained.

In the multilevel methods of modeling and solving a decision problem, a complete problem representation is put together from subproblems, where each subproblem refers to some part of the whole problem situation. The subproblems are to some extent independent of one another, but not totally—there are certain ties between them. That is, the total model complex is constructed out of a set of building blocks, each subproblem constituting one such block. The subproblems form an interrelated hierarchy, which means that they are considered to be on different hierarchical levels.

Hence, the total problem situation is modeled as consisting of a set of smaller subproblems. If these subproblems were now solved individually and the resulting subproblem solutions pieced together, one would not necessarily obtain a satisfactory solution to the overall decision problem. That is, it is not always realistic to subdivide a problem into smaller subproblems and then hope that completely independent solution of these subproblems will provide an acceptable solution to the overall problem. Rather, the subproblems must be coordinated in some fashion, and one function of higher-level subproblems is to coordinate the lower-level ones. For instance, in a two-level representation of a given decision problem, there may be three subproblems on the second (lowest) level, and one on the first. One function of the first-level subproblem is then to coordinate the second-level subproblems. We will be more specific later on about how coordination is achieved; at this point, we note only that coordination usually involves an iterative information exchange between levels.

One may now wonder why one would want to use a multilevel method for a given decision problem. Very briefly, there are at least three reasons. First, the given problem may be so large and complex that solution by conventional, single-level methods (such as ordinary linear programming) is not feasible—the number of variables, for example, may be too great. That is, the capacity of existing computing machinery could be exceeded. Second, it may be that the given problem can actually be solved by conventional, single-level methods, but the total modeling and problem-solving effort (including flow-charting, programming, card punching, debugging, and computer time) would be smaller using a multilevel method. Finally, a multilevel methodology allows for flexible and sophisticated modeling and problem solving. If one wants to solve the given problem directly, in a single-level fashion, one may be forced to use some rather crude method (like ordinary linear programming). Multilevel systems analysis allows one to use different techniques for handling different subproblems.

At this point, we shall not comment on the validity of these arguments in favour of multilevel systems analysis. We shall return to them, however, in the final chapter, where a more complete evaluation of multilevel systems analysis will be attempted.

In summary, multilevel systems analysis refers to a group of methods for modeling and solving decision problems. Common to these methods is that they involve a representation of the overall decision problem and a solution process in terms of subproblems on different levels, where lower-level subproblems are coordinated by higher-level ones. Instead of multilevel systems analysis, one often talks about hierarchical systems analysis, and the two words will be used interchangeably in this volume.

1.2 THE MULTILEVEL CHARACTER OF SYSTEMS

It is a trivial observation that there are all sorts of hierarchies and multilevel structures in the world around us. The literature dealing with hierarchical phenomena is large and is increasing rapidly. We cite only two areas as examples: within biology, there is widespread interest in multilevel structures, and one can, for instance, find discussions of hierarchical schemes such as the following: micromolecule—macromolecule—polymer—ultrastructural array—cell—tissue—organ—organism (Grene and Mendelsohn 1976, p. 149; Milsum 1972, p. 148; Pattee 1973, p. 5). In geography, one object of investigation is the hierarchy of cities, or central places, within an area (see, e.g., Beckman 1975). Additional examples could easily be cited. In fact, Kornai states: "Actually existing systems are multilevel" (Kornai 1971, p. 83).

However, in this volume we are not concerned with hierarchies in general. We are concerned with multilevel systems analysis methods for modeling and solving problems in economics and management. Our purpose is hence a normative one—to discuss how a set of multilevel techniques can be utilized as decision-making aids in certain decision problem situations, and we are not much interested in descriptive investigations of hierarchies in nature and society.

Nonetheless, the hierarchical nature of one kind of system—organizations—will be considered briefly, since the kinds of decision problems treated in this volume arise in organizations and have to be solved within an organizational context. Virtually all formal organizations are hierarchical in nature. In fact, organization theorists consider hierarchical structure to be one of the most important characteristics of an organization. Many organization-theoretic studies have investigated aspects of hierarchy, for example number of hierarchical levels from company president to worker on the factory floor, or average span of control (see, for instance, Blau 1968, Blau *et al.* 1966, Meyer 1972).

The multilevel structure of organizations is of interest to us because in some cases, a multilevel model of a given problem situation involves subproblems that may be thought of as "belonging to" different organizational subunits on different levels in the organization chart. For instance, in Chapter 6 we will

consider certain decision problems involving the planning of production and sales in corporations. The two highest hierarchical levels in a corporation are often the central leadership group (referred to here as headquarters) and functional departments. The production and sales planning problem can be represented in a multilevel fashion as a set of subproblems, one each for headquarters and the functional departments. The subproblem pertaining to a production department refers to planning production, that pertaining to a sales department refers to planning sales, and so on. The subproblem pertaining to headquarters is a higher-level one than the departmental subproblems, and one function of the headquarters subproblem is to coordinate the departmental ones. Thus, the hierarchy of subproblems in this case corresponds to the hierarchy of organizational subunits in the organization facing the decision problem.

In conclusion, it may be mentioned that modeling in a multilevel fashion is to a large extent a question of design. That is, a representation of the given problem situation is being built up in a hierarchical fashion from building blocks, with the subproblems constituting the blocks. In other design situations, such as architectural design or organization design, an analogous procedure is sometimes followed. That is, the total structure in those situations, too, is assembled in a hierarchical manner from smaller building blocks (see, e.g., Gerwin 1974).

1.3 THE HISTORICAL DEVELOPMENT OF MULTILEVEL SYSTEMS ANALYSIS

We do not intend here to trace out in detail the development of multilevel systems analysis, but three major sources of inspiration should be mentioned: the economic systems debate of the 1930s, the Dantzig–Wolfe decomposition principle, and hierarchical systems theory, as developed by Mesarovic and his associates.

Consider the problem of planning production in a socialized economy. Von Mises, a leading opponent of socialism at the time, argued that rational production planning would be impossible if all industries were nationalized, since there would be no free markets for raw materials, intermediate goods, and capital goods. There would hence be no prices, which are necessary as guides in arriving at rational production plans (von Mises 1935). Alternatively, von Hayek argued that there could theoretically exist such prices in a socialized economy, but it would be very difficult to find them (von Hayek 1935). Taylor (1939) and Lange (1939), on the other hand, argued that rational production planning would, indeed, be possible. Suppose we introduce a Central Planning Board into the socialized economy. The Central Planning Board would be responsible for price formation, as follows: Prices for the various commodities

in the economy are announced by the Central Planning Board. Consumers and production managers react to these prices by carrying out production activities and market transactions, taking the announced prices as given. If the price for some good is incorrectly specified, there will be a surplus or deficit of that good. The price must then be adjusted by the Central Planning Board (decreased or increased). That is, through a trial-and-error process one can find prices that equalize supply and demand of the various commodities, and those prices are the ones required for rational production planning.

This procedure is seen to be of a two-level type: The total problem of planning production in the economy is divided into subproblems, one for each plant manager, and one for the Central Planning Board. The plant manager subproblems, which are the second-level ones, are to plan production in the respective plants, taking the announced prices as given. The Central Planning Board subproblem, which is the first-level one, is to find the equilibrating commodity prices. In so doing, the Central Planning Board subproblem coordinates the plant manager subproblems. Much of the discussion by Lange (1939) and Taylor (1939) is, in fact, a verbal statement of one particular two-level method.

The economic systems debate of the 1930s reflects the influence of economic theory on the development of multilevel systems analysis. The Dantzig–Wolfe decomposition principle reflects the influence of operations research. One of the major breakthroughs in operations research was the development around 1947 of the simplex method for linear programming. It gradually became clear, however, that the ordinary simplex method is not ideally suited for very large linear programming (LP) problems or for problems with special structure. The first two-level method developed for LP problems with special structure is probably that of Ford and Fulkerson (1958), whose paper is an example of column generation (see Chapter 3 for a further discussion of this technique). The problem considered by Ford and Fulkerson is one of maximal multicommodity network flows. This problem may be formulated as a linear program, where each column represents one particular path from source to sink for one commodity. However, there may be many columns, and, rather than specifying them all in advance, Ford and Fulkerson suggest that they should be generated as needed, through a shortest-path algorithm. The Ford and Fulkerson paper is important historically in that it is a direct forerunner of the Dantzig–Wolfe decomposition principle, a fact also acknowledged by Dantzig himself (Dantzig 1963, p. 449).

The Dantzig–Wolfe decomposition principle (Dantzig and Wolfe 1961) was developed around 1960, sparking intensive research work on the development of various types of decomposition schemes for linear and nonlinear programming problems. Many of these schemes are, in fact, multilevel methods. Apart from whatever usefulness it may have as a purely computational tool (this matter is reviewed in Chapter 4), the Dantzig–Wolfe decomposition principle

has profoundly influenced the way economists, management scientists, and systems analysts think about multilevel methods (see, for instance Alekseev 1975, p. 13; Katsenelinboigen and Faerman 1967, p. 336).

Hierarchical systems theory is an abstract theory of how problems may be modeled and solved in a multilevel fashion. The most authoritative statement of this theory is contained in a book by Mesarovic *et al.* (1970), which summarizes research carried out by Mesarovic and his associates over several years. Hierarchical systems theory has received wide attention from, among others, systems-oriented organization theorists. It probes into the conceptual foundations of hierarchical problem solving, for instance, consistency and coordinability in a hierarchy of subproblems.

We have briefly described the historical development of multilevel systems analysis. We will see later how topics like the Dantzig–Wolfe decomposition principle and hierarchical systems theory provide theoretical underpinnings for much of the development in this book.

1.4 OVERVIEW OF THE VOLUME

This volume has eleven chapters. Chapter 2 contains some basic theory and concepts in multilevel systems analysis, and Chapter 3 presents multilevel techniques, largely falling within column generation and decomposition in mathematical programming. The selection of topics in Chapters 2 and 3 is not comprehensive in the sense that every imaginable multilevel concept or method is treated. Rather, the intention is to convey some basic ideas and to enable the reader to follow the rest of the volume. Chapter 4 discusses experiences in applying the Dantzig–Wolfe decomposition method for linear programming to different test problems. The Dantzig–Wolfe method is one of the methods discussed in Chapter 3. Several authors have carried out numerical experiments with that method, and those experiments are summarized in a separate chapter, Chapter 4.

A large part of the book is devoted to applications of multilevel methods to concrete decision problems (Chapters 5–10). These applications are drawn from various areas in economics and management, and they represent case studies in the actual usage of multilevel systems analysis. Chapter 5 considers national and regional economic planning. It discusses the use of multilevel methods for 5-year planning in Hungary and for national economic planning in Mexico; a regional planning model taken from a Soviet planning situation is also presented. Chapter 6 discusses experimental two-level methods for production and sales planning in two corporations (a Swedish paperboard manufacturer and a Danish slaughterhouse). In Chapter 7, dealing with operations management, two alternative multilevel approaches to production scheduling are presented. Chapter 8 considers two problem situations regard-

ing deliveries from producing factories to customers, and Chapter 9 takes up two more planning situations, which could not conveniently be covered in the other chapters: freight ship route scheduling and electricity generation. Chapter 10 discusses multilevel approaches to water pollution control.

Chapter 11 contains a concluding appraisal of multilevel systems analysis. It draws upon the case studies in the previous chapters and attempts to answer the question: How useful is the multilevel methodology in modeling and solving decision problems?

It is clear that a major portion of this volume is devoted to a number of case studies of the use of multilevel techniques in representative problem areas. Our interest is focused more on the multilevel methodology than on the substance of the problem areas as such. For instance, Chapter 7 is called "Operations Management," but that should not be interpreted to mean that we will give an encyclopedic treatment of that topic. Rather, we present only a few multilevel methods that can be used in operations management. In general, we have chosen our case studies somewhat opportunistically, picking out studies that are interesting as illustrations of multilevel ideas. As a consequence of our focus on the multilevel methodology, we do not discuss data-gathering problems very much. Some of the case studies presented represent a very substantial empirical research effort, with a great deal of work going into data collection. One could argue that the data-collection work is the really interesting part of those studies—but not from our point of view. Also, as was pointed out in the first section of this chapter, multilevel system analysis entails representing a given problem situation as a hierarchy of subproblems. We will not always be very specific about how the subproblems are to be solved (e.g., which particular nonlinear or dynamic programming algorithm is to be used). Rather, we will concentrate on the interrelationships between the subproblems. This also conforms with our focus on the multilevel methodology.

To qualify for inclusion in this volume, we require of our case studies that they deal with the real world at least to some extent. That is, we wish to exclude purely academic exercises. Thus, we have included only cases where multilevel methods have been tried out with real-world data. That does not necessarily mean that they have to have been implemented and applied on a regular, routine basis, but they should at least have been tried out in some experimental situations.

Considering further our criteria for selecting case studies and models for inclusion, we do not wish to include a model just because it deals with a phenomenon that could be considered multilevel in some general sense. To cite only one example, Davies (1976) describes a model for manpower planning. This model deals with a multilevel phenomenon, namely, manpower of different hierarchical grades. However, no multilevel methods are involved in the solution process. This appears to be multilevelness of a rather trivial kind, and no such models are discussed in this volume.

Windsor and Chow (1970) formulate a model that is multilevel in a somewhat less trivial sense. A problem in farm irrigation is solved in two steps. In the first step, an irrigation policy is determined by dynamic programming. In the second, linear programming is used to determine (among other things) the crop mix. One could now describe this arrangement as a multilevel configuration, where a dynamic programming subproblem is placed above a linear programming subproblem. Windsor and Chow do, in fact, refer to their approach as two-level. However, we consider it an instance of ordinary sequential decision-making, and to include their study among our cases would imply that essentially all sequential decision problems could be regarded as multilevel. For that reason, we have included mainly case studies where there is at least some rudimentary *iterative interaction between the subproblem levels.* For instance, suppose the multilevel representation of a given problem situation involves two levels, with three subproblems on the second level and one on the first. The first-level subproblem then coordinates the second-level ones. For the resulting two-level model to qualify for inclusion in this volume, we require some iterative interaction between the first-level subproblem and the second-level ones. There is no such iterative interaction in the Windsor–Chow example. We have made only one exception to this rule—we have included one model of the Windsor–Chow type in Chapter 7 on operations management. Our reasons for doing so are stated in that chapter.

In fairness to the reader, we wish to emphasize at the outset that this volume does not present a complete picture of multilevel systems analysis. In fact, the presentation is largely limited to multilevel systems analysis methods founded on linear programming and immediate extensions of linear programming. In particular, we do not include control-theoretic model formulations in this volume. It may be that the most convincing applications of multilevel systems analysis are, in fact, to control-type problems, including on-line control of ongoing processes with feedback loops. Nevertheless, such problems are not considered here. We remain within the realm of one-shot, discrete-time decision making—deciding, for example, on a production plan (or a shipping plan, a 5-year plan, or a regional development plan) for the next planning period. Our reason for not including control-theoretic models is that we want to minimize the mathematical apparatus required. However, for completeness we have added one brief section in Chapter 3 on multilevel approaches to control-type problems.

Our notation in the following chapters is straightforward and conventional. Equations and other expressions are numbered consecutively in each chapter. Optimization problems are sometimes written in maximization, other times in minimization format. We also state restrictions sometimes as equalities, other times as inequalities, depending on what is most natural in each situation.

REFERENCES

Alekseev, A. M. 1975. Mnogourovnevye sistemy planirovaniia promyshlennogo proizvodstva. (Multilevel Systems for Planning Industrial Production, in Russian.) Novosibirsk: Nauka (Siberian department).

Beckman, M. J. 1975. On the economic structure of strictly hierarchical central place systems. Environment and Planning A 7(7): 815–820.

Blau, P. M. 1968. The hierarchy of authority in organizations. American Journal of Sociology 73: 453–467.

Blau, P. M., W. V. Heydebrand, and R. E. Stauffer. 1966. The structure of small bureaucracies. American Sociological Review 31: 179–191.

Dantzig, G. B. 1963. Linear Programming and Extensions. Princeton, New Jersey: Princeton University Press.

Dantzig, G. B., and P. Wolfe. 1961. The decomposition algorithm for linear programs. Econometrica 29: 767–778.

Davies, G. S. 1976. Consistent recruitment in a graded manpower system. Management Science 22: 1215–1220.

Ford, L. R., and D. R. Fulkerson. 1958. Suggested computation for maximal multi-commodity network flows. Management Science 5: 97–101.

Gerwin, D. 1974. A Systems Framework for Organizational Structural Design. Report 1/74–32. Berlin: International Institute of Management.

Grene, M., and E. Mendelsohn (ed.). 1976. Topics in the Philosophy of Biology. (Synthese Library, Vol. 84.) Dordrecht, Holland: Reidel.

von Hayek, F. A. 1935. The present state of the debate, pp. 201–243. In F. A. von Hayek (ed.), Collectivist Economic Planning. London: Routledge.

Katsenelinboigen, A. I., and E. Iu. Faerman. 1967. Centralism and economic independence in the socialist economy. (In Russian.) Ekonomika i matematicheskie metody 3(3): 331–346.

Kornai, J. 1971. Anti-Equilibrium. Amsterdam: North-Holland.

Lange, O. 1939. On the economic theory of socialism, pp. 55–143. In B. E. Lippincott (ed.), On the Economic Theory of Socialism. Minneapolis: University of Minnesota Press.

Mesarovic, M. D., D. Macko, and Y. Takahara. 1970. Theory of Hierarchical, Multilevel Systems. New York: Academic.

Meyer, M. W. 1972. Bureaucratic Structure and Authority. New York: Harper and Row.

Milsum, J. H. 1972. The hierarchical basis for general living systems, pp. 145–187. In G. J. Klir (ed.), Trends in General Systems Theory. New York: Wiley.

von Mises, L. 1935. Economic calculation in the socialist commonwealth, pp. 87–130. In F. A. von Hayek (ed.), Collectivist Economic Planning. London: Routledge.

Pattee, H. H. (ed.). 1973. Hierarchy Theory. New York: George Braziller.

Taylor, F. M. 1939. The guidance of production in a socialist state, pp. 39–54. In B. E. Lippincott (ed.). On the Economic Theory of Socialism. Minneapolis: University of Minnesota Press.

Windsor, J. S., and V. T. Chow. 1970. A Programming Model for Farm Irrigation Systems. Report No. 23. Hydraulic Engineering Series, Department of Civil Engineering. Urbana, Illinois: University of Illinois at Urbana-Champaign.

2 Fundamental Concepts

2.1 THE IDEALIZED MULTILEVEL APPROACH

2.1.1 PARTITIONING THE OVERALL PROBLEM INTO SUBPROBLEMS

Multilevel systems analysis is not one precisely defined technique. Rather, it may be described as an *approach* with several variants. In this section, we will consider one particular variant, which could be labeled idealized. It is idealized because it employs only two levels in the subproblem hierarchy. It is also idealized in some other respects, as will become clear later, in section 2.2, where we discuss more general forms of multilevel systems analysis. The discussion in this section draws on Jennergren (1976) and Mesarovic *et al.* (1970, pp. 85–106).

Our purpose here is to present some of the most basic ideas in multilevel systems analysis. In particular, we will discuss how a given overall problem may be partitioned into supremal and infimal subproblems. Some set-theoretic notation will be used. This may seem abstract at first, but concrete examples are given in the following section.

Consider some given decision problem, which is denoted by $\mathscr{D}$. That is, $\mathscr{D}$ is a generic symbol for some problem. $\mathscr{D}$ is not identical with the underlying real-world decision situation; rather, some preliminary modeling has already been done in order to arrive at $\mathscr{D}$. In almost all the case studies described later in this volume, $\mathscr{D}$ is stated as an optimization problem of the mathematical programming type.* $\mathscr{D}$ is referred to as the *overall*, or *original*, problem. For the present discussion, $\mathscr{D}$ may be written as follows:

$$\text{Out of all } m \in M, \quad \text{find one for which } m \in \bar{M}.$$

* This should not be taken to mean that in all the case studies, an optimal solution to $\mathscr{D}$ is obtained. Often it is not, and, indeed, it is never intended or expected that an optimal solution will be obtained. The optimization formulation is merely a convenient starting point.

M is the feasible set, and $\bar{M}$ the acceptable set. Any $m \in (M \cap \bar{M})$ is a solution to $\mathscr{D}$. This specification of $\mathscr{D}$ is quite general and includes not only optimization problems. In many cases, $\mathscr{D}$ can probably be solved directly (e.g., by ordinary linear programming), and there is hence no need to use a multilevel methodology.

However, to continue with the idealized multilevel approach, suppose $\mathscr{D}$ is partitioned into subproblems as follows: Let there be n subproblems on the lowest level, referred to as *infimal subproblems*, and denoted $\mathscr{D}_1(\gamma) \ldots \mathscr{D}_n(\gamma)$. Let there be one subproblem $\mathscr{D}_0$ on the top level, the *supremal subproblem.* Each infimal subproblem $\mathscr{D}_j(\gamma)$ is specified as

$$\text{Out of all } x_j \in X_j^\gamma, \quad \text{find one for which } x_j \in (X_j^\gamma \cap \bar{X}_j^\gamma).$$

Here, X_j^γ is the feasible set and $\bar{X}_j^\gamma$ the acceptable set. The notation X_j^γ and $\bar{X}_j^\gamma$ indicates that the feasible set, and the acceptable set, and hence the infimal subproblem as a whole, depends on the parameter γ. γ could, for instance, be a vector of real numbers. The parameter γ is specified by the supremal subproblem and is the means of coordinating the infimal subproblems. The fundamental assumption underlying the idealized multilevel approach is that a solution to the original problem $\mathscr{D}$ can be obtained from solutions to the infimal subproblems $\mathscr{D}_1(\gamma) \ldots \mathscr{D}_n(\gamma)$. For simplicity of notation, let $\bar{\mathscr{D}}(\gamma) = (\mathscr{D}_1(\gamma) \ldots \mathscr{D}_n(\gamma))$ and $x = (x_1 \ldots x_n)$. To transform a solution x to $\bar{\mathscr{D}}(\gamma)$ (i.e., $x \in (X_1^\gamma \cap \bar{X}_1^\gamma) \times (X_2^\gamma \cap \bar{X}_2^\gamma) \times \cdots \times (X_n^\gamma \cap \bar{X}_n^\gamma)$) into a solution to $\mathscr{D}$, one needs a mapping π_M: $x \to m$, i.e., $m = \pi_M(x)$. In other words, π_M is a mechanism that states how a candidate solution to $\mathscr{D}$ is to be composed from solutions to the infimal subproblems $\bar{\mathscr{D}}(\gamma)$. However, it is not to be expected that any collection of infimal subproblems $\bar{\mathscr{D}}(\gamma)$, arbitrarily specified, will result in a solution to $\mathscr{D}$, by way of the mapping π_M. For that purpose, the infimal subproblems must be coordinated by the supremal subproblem. The supremal subproblem can hence be stated as:

Find γ such that

1. $(X_j^\gamma \cap \bar{X}_j^\gamma) \neq \varnothing$ for $j = 1 \ldots n$
2. $\pi_M(x) \in (M \cap \bar{M})$ for any $x \in (X_1^\gamma \cap \bar{X}_1^\gamma) \times (X_2^\gamma \cap \bar{X}_2^\gamma) \times \cdots \times (X_n^\gamma \cap \bar{X}_n^\gamma)$

Stated in words, the parameter γ must be such that, in the first place, each infimal subproblem has a solution; and, in the second place, any candidate solution to the overall problem put together from infimal subproblem solutions actually does the job—that is, it solves the overall problem.

The collection $(\mathscr{D}_0, \bar{\mathscr{D}}(\gamma))$ is the result of the partition of the original problem $\mathscr{D}$ into a two-level hierarchy of subproblems. That is, $(\mathscr{D}_0, \bar{\mathscr{D}}(\gamma))$ is a two-level representation of the original problem $\mathscr{D}$. If there exists at least one γ satisfying conditions (1) and (2) in the supremal subproblem $\mathscr{D}_0$, then the two-level

subproblem hierarchy is *coordinable* relative to the original problem $\mathscr{D}$. If the two-level subproblem hierarchy is not coordinable relative to $\mathscr{D}$, it could be because the subproblem hierarchy is badly designed; i.e., a different subproblem hierarchy could be coordinable relative to $\mathscr{D}$. However, there are two other cases where coordinability does not hold. If $M = \emptyset$ or $\bar{M} = \emptyset$, in both cases implying that $(M \cap \bar{M}) = \emptyset$, then it is easy to see that no subproblem hierarchy, however designed, can be coordinable relative to $\mathscr{D}$. This implies that $\mathscr{D}$ is probably badly specified to begin with. (On the concept of coordinability, see also Jennergren 1974.)

Let us take stock of what has been achieved: The overall problem $\mathscr{D}$ has been partitioned into a two-level hierarchy of subproblems, with $\mathscr{D}_0$ being the supremal subproblem and $\mathscr{D}_1(\gamma) \ldots \mathscr{D}_n(\gamma)$ being the infimal ones. Alternatively, $\mathscr{D}_0$ and $\mathscr{D}_1(\gamma) \ldots \mathscr{D}_n(\gamma)$ form a two-level representation of $\mathscr{D}$, assembled from $(n+1)$ building blocks, each subproblem being one of those blocks. If the two-level representation $\mathscr{D}_0$, $\mathscr{D}_1(\gamma) \cdots \mathscr{D}_n(\gamma)$ is coordinable relative to $\mathscr{D}$, then $\mathscr{D}$ and $(\mathscr{D}_0, \bar{\mathscr{D}}(\gamma))$ are equivalent. In that case, a solution to $\mathscr{D}$ may be obtained by considering the two-level hierarchy $(\mathscr{D}_0, \bar{\mathscr{D}}(\gamma))$.

2.1.2 EXAMPLES OF TWO-LEVEL SUBPROBLEM HIERARCHIES

To illustrate the ideas in the preceding section, some examples will be given. Consider first an LP problem with special structure:

$$
\begin{aligned}
\text{Maximize} \quad & c_1x_1 + c_2x_2 + \cdots + c_nx_n \\
\text{s.t.:} \quad & A_1x_1 + A_2x_2 + \cdots + A_nx_n \leq a, \\
& B_1x_1 \leq b_1, \\
& B_2x_2 \leq b_2, \\
& \qquad \cdots \\
& B_nx_n \leq b_n,
\end{aligned}
\tag{2.1}
$$

$$x_1, x_2 \ldots x_n \geq 0.$$

A_i and B_i are constant matrices, c_i, b_i, and a constant vectors, and x_i variable vectors of suitable dimensions.* Problem (2.1) has a *decomposable* or *block-angular* structure. The constraints $A_1x_1 + A_2x_2 + \cdots + A_nx_n \leq a$ are referred to as *coupling* constraints. Let X denote the set of feasible solutions to (2.1) and $\bar{X}$ the set of optimal solutions. Problem (2.1) is the overall problem $\mathscr{D}$. It will now be shown how the overall problem (2.1) may be partitioned into a two-level subproblem hierarchy.

* In this example we use x rather than m to denote the variables in the overall problem, to conform with usual LP notation.

A first two-level representation is obtained by specifying infimal subproblems as follows. For each index $j = 1 \ldots n$:

$$\begin{aligned} &\text{Maximize} && c_j x_j \\ &\text{s.t.:} && A_j x_j \leq a_j, \\ & && B_j x_j \leq b_j, \\ & && x_j \geq 0. \end{aligned} \tag{2.2}$$

The problem formulations (2.2) are the infimal subproblems $\mathscr{D}_j(\gamma)$. The parameter γ is in this case $(a_1, a_2 \ldots a_n)$, and it is seen that (2.2) depends only on the jth component of $(a_1, a_2 \ldots a_n)$, a_j. The vector a_j has, of course, the same decision as a. In (2.2), the feasible set depends on a_j, but the objective function does not. Let $X_j(a_j)$ denote the set of feasible solutions to (2) and $\bar{X}_j(a_j)$ the set of optimal solutions. Obtaining a solution to (2.1) from the infimal subproblems (2.2) is easy in this case. One merely puts all the x_j-vectors together: $(x_1, x_2 \ldots x_n)$. That is, the mapping π_M is trivial to specify.

The supremal subproblem $\mathscr{D}_0$ is

Find $(a_1, a_2 \ldots a_n)$ such that

1. $(X_j(a_j) \cap \bar{X}_j(a_j)) \neq \varnothing \ (j = 1 \ldots n)$
2. $(x_1, x_2 \ldots x_n) \in (X \cap \bar{X})$ for any $(x_1, x_2 \ldots x_n) \in$
   $[X_1(a_1) \cap \bar{X}_1(a_1)] \times [(X_2(a_2) \cap \bar{X}_2(a_2)] \times \cdots \times [X_n(a_n) \cap \bar{X}_n(a_n)]$

It is easy to show that if (2.1) has an optimal solution (i.e., $(X \cap \bar{X}) \neq \varnothing$), the two-level hierarchy is coordinable with respect to the original problem (2.1). Without loss of optimality, one may restrict one's attention to $a_1, a_2 \ldots a_n$ such that $\Sigma a_j = a$ in solving the supremal subproblem. What this two-level hierarchy achieves, then, is to partition the right-hand side of the coupling constraints of (2.1) among the infimal subproblems.

Consider now another two-level representation of the original problem (2.1). Let the infimal subproblem for each index j be written

$$\begin{aligned} &\text{Maximize } c_j x_j - p A_j x_j \\ &\text{s.t.:} \quad B_j x_j \leq b_j, \\ &\qquad\quad x_j \geq 0. \end{aligned} \tag{2.3}$$

In this case, the parameter γ in the specification of the infimal subproblem $\mathscr{D}_j(\gamma)$ is p, which may be thought of as a price vector associated with the coupling constraints of (2.1). Let $X_j(p)$ denote the set of feasible solutions to (2.3), and $\bar{X}_j(p)$ the set of optimal solutions. In this case, $X_j(p)$ does not actually depend on p; the notation is used for consistency. Recovering a

candidate solution to (2.1) from solutions to each infimal subproblem is easy in this case, too; one merely puts all the x_j-vectors together, as in the previous case.

The supremal subproblem $\mathscr{D}_0$ is

Find p such that

1. $(X_j(p) \cap \bar{X}_j(p)) \neq \varnothing$ $(j = 1 \ldots n)$
2. $(x_1, x_2 \ldots x_n) \in (X \cap \bar{X}$ for any $(x_1, x_2 \ldots x_n) \in$
   $[X_1(p) \cap \bar{X}_1(p)] \times [X_2(p) \cap \bar{X}_2(p)] \times \cdots \times [X_n(p) \cap \bar{X}_n(p)]$

One can now show that if (2.1) has a unique, nondegenerate optimal solution, then there exists *no solution* to $\mathscr{D}_0$ (the situation where none of the coupling constraints is binding at the optimum is a trivial exception). If at least one coupling constraint is binding, this two-level hierarchy of subproblems is hence not coordinable with respect to the overall problem (2.1) (except in certain degenerate situations ruled out if (2.1) has a unique, nondegenerate optimal solution). This is a well-known fact that has been discussed by several authors in the management science literature (see, e.g., Baumol and Fabian 1964). If the optimal solution to (2.1) is unique and nondegenerate, the optimal dual multiplier vector is also unique. Suppose one sets the parameter p used for coordinating the infimal subproblems (2.3) equal to the optimal dual multipliers associated with the coupling constraints. Even with that specification of p, a candidate solution to (2.1) assembled from infimal subproblem solutions will usually be either infeasible for (2.1) or feasible but nonoptimal. Consider, for instance, the following simple LP problem:

$$
\begin{aligned}
\text{Maximize} \quad & x_{11} + x_{21} + 2x_{12} + x_{22} \\
\text{s.t.:} \quad & x_{11} + 2x_{21} + 2x_{12} + x_{22} \leq 40, \\
& x_{11} + 3x_{21} \leq 30, \\
& 2x_{11} + x_{21} \leq 20, \\
& x_{12} \leq 10, \\
& x_{22} \leq 10, \\
& x_{12} + x_{22} \leq 15, \\
& x_{11}, x_{21}, x_{12}, x_{22} \geq 0.
\end{aligned}
$$

In this case, (25/3, 10/3, 10, 5) is the unique and nondegenerate optimal solution. The optimal dual multiplier associated with the coupling constraint

$x_{11}+2x_{21}+2x_{12}+x_{22}\leq 40$ is 1/3. Suppose the following two infimal subproblems, of the same type as (2.3), are constructed:

$$\begin{aligned}
&\text{Maximize} && x_{11}+\ x_{21}-(1/3)(x_{11}+2x_{21})\\
&\text{s.t.:} && x_{11}+3x_{21} && \leq 30,\\
& && 2x_{11}+\ x_{21} && \leq 20,\\
& && x_{11},\ x_{21} && \geq 0,\\
&\text{Maximize} && 2x_{12}+x_{22}-(1/3)(2x_{12}+x_{22})\\
&\text{s.t.:} && x_{12} && \leq 10,\\
& && x_{22} && \leq 10,\\
& && x_{12}+x_{22} && \leq 15,\\
& && x_{12},x_{22} && \geq 0.
\end{aligned}$$

The first of these two infimal subproblems has infinitely many optimal solutions, of which (25/3, 10/3) is merely one. This illustrates the noncoordinability phenomenon. In fact, noncoordinability holds for any choice of p, as is easily verified for this example. It should be mentioned, though, that for certain *nonlinear* problems, coordinability may hold (see section 3.7).

Consider now a third two-level representation of the overall problem (2.1), which may be thought of as an extension of the immediately preceding one. Instead of coordinating the infimal subproblems by prices, suppose one uses price schedules of the type $(r+kx_j^TA_j^T)$ (the superscript T denotes transpose). That is, the prices associated with the coupling constraints now depend on the extent to which those constraints are utilized in the individual infimal subproblem. Each infimal subproblem then becomes

$$\begin{aligned}
&\text{Maximize} && c_jx_j-(r+kx_j^TA_j^T)A_jx_j\\
&\text{s.t.:} && B_jx_j\leq b_j,\\
& && x_j\geq 0.
\end{aligned}\tag{2.4}$$

The parameter γ here corresponds to the vector r and the constant k. Let the set of feasible solutions to (2.4) be denoted $X_j(r, k)$ (which does not depend on r and k), and the set of optimal solutions $\bar{X}_j(r, k)$. The supremal subproblem is now

Find r and k such that

1. $(X_j(r, k)\cap\bar{X}_j(r, k))\neq\varnothing$ $(j=1\ldots n)$
2. $(x_1, x_2\ldots x_n)\in(X\cap\bar{X})$ for any $(x_1, x_2\ldots x_n)\in[X_1(r, k)\cap\bar{X}_1(r, k)]\times[(X_2(r, k)\cap\bar{X}_2(r, k)]\times\cdots\times[X_n(r, k)\cap\bar{X}_n(r, k)]$

If $(X \cap \bar{X}) \neq \varnothing$, the supremal subproblem has a solution. That is, this third subproblem hierarchy is coordinable with respect to (2.1) (Jennergren 1973). Apparently, the infimal subproblems may now be thought of as coordinated by means of nonconstant prices (price schedules) associated with the coupling constraints.

A different overall problem will now be taken up. It may be referred to as "the transfer price problem" (Hirshleifer 1956). A company has two departments, or factories, the production program of each of which consists of one product. Department 1 transfers part of its product to department 2 for further refinement and marketing there, the remainder being sold to outside buyers. One unit of the output of department 1 is needed for every unit of output of department 2. Apart from this interdependence, there is technological and market independence between the two departments. Let

m_{11}: the number of units produced by department 1 and sold outside buyers

m_{21}: the number of units produced by department 1 and transferred to department 2

m_2: the number of units produced by department 2 (all sold to outside buyers)

$R_1(m_{11})$: the total revenue obtained by selling m_{11} units of the output of department 1 to outsiders

$R_2(m_2)$: the total revenue obtained by selling m_2 units of the output of department 2 to outsiders

$C_1(m_{11}, m_{21})$: the total cost associated with producing $(m_{11}+m_{21})$ units in department 1

$C_2(m_2)$: the total cost associated with producing m_2 units in department 2

The overall problem $\mathscr{D}$ is one of selecting the production levels in both departments that maximize total company profit. This may be written as:

$$\begin{aligned} &\text{Maximize} && R_1(m_{11})+R_2(m_2)-C_1(m_{11}, m_{21})-C_2(m_2) \\ &\text{s.t.:} && m_{21}=m_2, \\ & && m_{11}, m_{21}, m_2 \geq 0. \end{aligned} \tag{2.5}$$

The restriction $m_{21}=m_2$ expresses the fact that one unit of the output of department 1 is needed as input to produce one unit in department 2. Let M and $\bar{M}$ be the feasible and optimal sets. In this situation, it is natural to associate one infimal subproblem each with department 1 and department 2. The supremal subproblem is associated with company headquarters. This is an instance of the situation referred to in section 1.2, where the subproblems in a multilevel model "belong to" different subunits of some organization. The

infimal subproblem of department 1 could be written as

$$\begin{aligned} &\text{Maximize} && R_1(x_{11}) + px_{21} - C_1(x_{11}, x_{21}) \\ &\text{s.t.:} && x_{11}, x_{21} \geq 0 \end{aligned}$$

and that of department 2 as

$$\begin{aligned} &\text{Maximize} && R_2(x_2) - px_2 - C_2(x_2) \\ &\text{s.t.:} && x_2 \geq 0. \end{aligned}$$

Let X_j^p and $\bar{X}_j^p$ denote the feasible and optimal sets, respectively ($j = 1, 2$). It is noted that X_j^p does not actually depend on p. Recovering a candidate solution to the original problem (2.5) from the infimal subproblems is easy: $(m_{11}, m_{21}, m_2) = (x_{11}, x_{21}, x_2)$.

The supremal subproblem is

Find p such that

1. $(X_j^p \cap \bar{X}_j^p) \neq \varnothing (j = 1, 2)$
2. For any $(x_{11}, x_{21}, x_2) \in (X_1^p \cap \bar{X}_1^p) \times (X_2^p \cap \bar{X}_2^p)$, $(m_{11}, m_{21}, m_2) = (x_{11}, x_{21}, x_2) \in (M \cap \bar{M})$

The parameter p, which is used to coordinate the infimal subproblems, may be interpreted as a transfer price. Depending on the properties of the revenue and cost functions, this two-level model of the transfer pricing situation may or may not be coordinable relative to the original problem (2.5).

For additional examples of two-level models of optimization problem situations, see Dirickx *et al.* (1973).

2.1.3 THE MULTILEVEL SOLUTION PROCESS

We have seen in the preceding two sections how, starting from a given overall problem $\mathscr{D}$, a two-level subproblem hierarchy, or simply a two-level model, of that decision problem is constructed. That hierarchy is composed of a set of infimal subproblems, $\mathscr{D}_1(\gamma) \ldots \mathscr{D}_n(\gamma)$, and a supremal subproblem, $\mathscr{D}_0$, which consists in finding those parameter values for which a candidate solution recovered from the infimal subproblems is also a solution to the original problem. However, finding those parameter values is usually not a trivial task. It usually requires certain computations. From this we infer that the two-level solution process actually involves two distinct phases, the *adjustment phase* and the *execution phase*. In the adjustment phase, a solution to the supremal subproblem is sought. In this phase, different two-level solution methods may be used, such as the Dantzig–Wolfe decomposition method (two-level solution methods are the topic of Chapter 3). The adjustment phase comes to a close

when a solution γ' to the supremal subproblem has been found. At that point, the execution phase starts. The execution phase consists of solving the infimal subproblems, using as inputs those parameter values γ' that were found during the adjustment phase. The overall solution is then recovered from the infimal subproblem solutions and executed as an actual decision, or plan.

The adjustment phase usually involves an iterative information exchange between supremal and infimal subproblems. These iterations customarily are carried out as follows: The infimal subproblems are solved, taking some current tentative parameter values as inputs. On the basis of the resulting infimal subproblem solutions, the tentative parameter values are revised (that is, the supremal subproblem is solved tentatively). This revision marks the beginning of the next iteration. The infimal subproblems are then solved again, using the revised parameter values as inputs, and so on. It is clear that this process may, indeed, be viewed as an iterative information exchange between supremal and infimal subproblems.

In most cases, the infimal subproblems will be specified in the same way in the adjustment phase as in the execution phase. However, this is not always so. Consider again the overall problem (2.1) (the decomposable LP problem) of the preceding section. Suppose one wishes to represent that problem in terms of a two-level subproblem hierarchy with infimal subproblems of the type (2.2). Then the supremal subproblem is one of finding a partition $(a_1, a_2 \ldots a_n)$ of the right-hand side a of the coupling constraints, such that the infimal subproblems together yield a solution to the overall problem (2.1). In the execution phase, the infimal subproblems will hence be of the type (2.2). However, if the Dantzig–Wolfe decomposition method is used to solve the supremal subproblem in the adjustment phase, the infimal subproblems in that phase will be of the type (2.3), not the type (2.2) (see the discussion of the Dantzig–Wolfe method in Chapter 3). In such cases, it would be most correct to talk about *two* different subproblem hierarchies being utilized, one during the adjustment phase, and one during the execution phase.

From the discussion in this section, it may be seen that the idealized two-level approach involves the following four steps in solving a problem:

1. Formalize, at least in a rough fashion, $\mathscr{D}$ (i.e., the overall decision problem under consideration). Identify natural subcomponents as candidates for infimal subproblems.

2. Partition $\mathscr{D}$ into a two-level subproblem hierarchy $(\mathscr{D}_0, \mathscr{D}_1(\gamma) \ldots \mathscr{D}_n(\gamma))$. Or, stated alternatively: construct a two-level model of $\mathscr{D}$ using infimal subproblems and a supremal subproblem as building blocks. The method that one intends to use to solve the supremal subproblem in the adjustment phase should be taken into consideration when this is done.

3. Determine whether a solution exists to the supremal subproblem—that is, whether the given two-level hierarchy of subproblems is coordinable

relative to $\mathscr{D}$. If not, and assuming that $(M \cap \bar{M}) \neq \varnothing$, the infimal subproblems must be modified, or redesigned, until coordinability holds.

4. Solve the supremal subproblem $\mathscr{D}_0$ by some suitable method. This is done in the adjustment phase and usually involves an iterative information exchange between supremal and infimal subproblems. When the supremal subproblem has been solved, the execution phase commences. The infimal subproblems $\mathscr{D}_1(\gamma) \ldots \mathscr{D}_n(\gamma)$ are solved, taking as inputs the parameter values obtained in the adjustment phase. The infimal subproblem solutions together now yield a solution to the overall problem $\mathscr{D}$.

If $(M \cap \bar{M}) = \varnothing$, then this would usually be discovered during the adjustment phase. If $\mathscr{D}$ is, for example, an LP problem where only optimal solutions are acceptable, then $(M \cap \bar{M}) = \varnothing$ comes about if there are no feasible solutions or feasible solution values tending toward (plus or minus) infinity. In both cases, the original problem $\mathscr{D}$ was probably wrongly specified at the outset.

2.2 ADDITIONAL ASPECTS OF MULTILEVEL SYSTEMS ANALYSIS

2.2.1 A MORE GENERAL MULTILEVEL APPROACH

In the previous section, a discussion of an "idealized" multilevel approach was given. For purposes of exposition, it was convenient to start out with that idealized approach. Some variants, or generalizations, will be discussed in this subsection.

The discussion so far has assumed that the solution to the overall problem $\mathscr{D}$ is implicit in the solutions to the infimal subproblems. That is, the role of the supremal subproblem, as described up to now, has been merely to coordinate the infimal subproblems. In the idealized multilevel approach, the adjustment phase is devoted to solving the supremal subproblem, but in the execution phase (i.e., after those parameter values have been found that do the coordination job) it is the infimal subproblem solutions that provide the solution to the overall problem. This is not always so; in some cases the resulting over-all problem solution is actually constructed from both infimal *and* supremal subproblem solutions, or even from the supremal subproblem solution alone. We shall see examples of all three situations in the following chapters.

In all three situations, the dichotomy between adjustment phase and execution phase remains; it is the mechanism for recovering the final solution to the original problem in the execution phase that differs. For instance, consider an overall problem $\mathscr{D}$ of the LP type but with many columns, where one wishes to use column generation. The supremal subproblem is then of the same type as

the original problem, but with only a subset of all the possible columns included. There will be one or several infimal subproblems to generate columns as needed. In the adjustment phase, information is exchanged between the supremal subproblem and the infimal ones, meaning in particular that one column for the supremal subproblem is obtained from each infimal subproblem in each iteration. However, in the execution phase, the ultimate solution is obtained from the supremal subproblem alone. In this situation, the role of the supremal subproblem consists of both coordinating the infimal subproblems (to obtain suitable new columns in each iteration of the adjustment phase) and providing the solution to the overall problem.

The coordinability condition defined in section 2.1.1 can be generalized to state that coordinability holds if the two-level subproblem hierarchy used in the adjustment phase is designed so that any candidate solution to the supremal subproblem in the execution phase is also a solution to the overall problem. Similarly, the coordinability condition can be generalized for the case where the ultimate solution to the original problem is derived from supremal and infimal subproblems jointly.

It should also be noted that the ensemble $\mathscr{D}_0$ and $\mathscr{D}_1(\gamma) \ldots \mathscr{D}_n(\gamma)$, i.e., the multilevel problem representation or model utilized, is not necessarily always equivalent to the overall problem $\mathscr{D}$. That is, coordinability does not always hold. By proceeding with a two-level problem representation that is not coordinable relative to the overall problem one hopes nevertheless to obtain a "reasonably good" solution to that problem. In section 7.2 we will see an instance of this.

A related variant of multilevel systems analysis is the following: Even if coordinability holds formally, this may in itself be unimportant, since approximate, or reasonably good, solutions to the overall problem may be all that is really sought. In several of the case studies reported later in this volume, the overall problem $\mathscr{D}$ is stated as an optimization problem, and a two-level problem representation is formulated that in itself is coordinable with respect to $\mathscr{D}$. However, this coordinability condition is not made use of, because the two-level solution method is terminated before optimality is reached.

There is yet another related variant: Sometimes there is no clear statement of the original problem $\mathscr{D}$ at all. That is, one starts out directly with the building blocks of a multilevel model, i.e., $\mathscr{D}_0$ and $\mathscr{D}_1(\gamma) \ldots \mathscr{D}_n(\gamma)$, without first specifying the overall problem $\mathscr{D}$, except possibly in a very loose manner. This is sometimes referred to as the *compositional* approach, as opposed to the *decompositional* approach, where one starts out with a description of the overall problem $\mathscr{D}$, and then constructs a multilevel subproblem hierarchy equivalent to $\mathscr{D}$ (see Sweeney *et al.* 1978). In Chapter 5, we will see an instance of the compositional approach, in the discussion of multilevel national

economic planning in Mexico. The overall problem may be stated as one of finding a good long-range development plan for the entire economy, but a formulation more specific that that is not given. In such a case, it is, of course, impossible to tell whether coordinability formally holds or not.

Characteristic for this group of related variants of multilevel systems analysis is that formal coordinability is unimportant, since a multilevel methodology is being used only to obtain an approximate solution to the overall problem.

Finally, only two-level subproblem hierarchies have been discussed so far. In fact, most applications of multilevel systems analysis to date have been of the two-level type, and this is reflected in the contents of this volume. However, a few three-level models are included here (in particular, a model of hierarchical production planning in Chapter 7 and freight ship scheduling in Chapter 9). When a three-level subproblem hierarchy is used, the procedure is still carried out in an adjustment phase and an execution phase. In the execution phase, the resulting solution to the overall problem is derived from subproblem solutions on one or more of the three levels in the hierarchy. In the adjustment phase, information is exchanged between the supremal subproblem and the intermediate ones, and between the intermediate and infimal ones. However, this information exchange can be carried out in different ways. For instance, how many iterations should there be between the intermediate and infimal levels for each supremal–intermediate iteration?

As for terminology, when there are three levels in the subproblem hierarchy, we will sometimes refer to the supremal subproblem level as No. 1 and the infimal subproblem level as No. 3. In general, the highest (supremal subproblem) level is denoted No. 1 throughout this volume.

In section 1.1, we gave a very brief description of what we mean by multilevel systems analysis. We may now be a bit more specific about it. Multilevel systems analysis is a common name for a group of approaches or methods for modeling and solving decision problems. All these methods construct a representation of the overall decision problem in terms of a hierarchy of subproblems. Since the subproblems are not totally independent of each other, they must be coordinated. Lower level subproblems are coordinated through parameters obtained from higher level ones. Also, since coordination cannot be achieved as a one-shot affair, there is an iterative procedure whereby the subproblems are solved several times. During the iterations of this procedure, referred to as the adjustment phase, trial coordinating parameter values are supplied to the lower level subproblems from higher level ones, and certain summary information travels in the other direction. In the execution phase, the resulting solution to the overall problem is derived from the solutions to subproblems on one or several levels. It is not excluded that the subproblem hierarchy, i.e., certain subproblems, may be specified differently in the two phases.

2.2.2 INSTITUTIONAL INTERPRETATIONS AND ANALOGIES

It was mentioned in section 1.2 that in some problem situations, a multilevel subproblem hierarchy corresponds to an organizational hierarchy. That is, the various subproblems may be thought of as belonging to different organizational subunits. The transfer pricing problem mentioned in section 2.1.2 is an example of an organizational analogy of this kind.

Looking into this matter of institutional analogies in more detail, we can distinguish at least three situations:

1. The subproblems in the multilevel representation do not correspond to organizational subunits in the real world. Hence, the subproblem hierarchy has no meaningful institutional interpretation. The maximal multicommodity network flow problem is one such case. The two-level representation of this problem involves infimal subproblems for generating columns (paths) for the supremal subproblem, but there is no meaningful correspondence between the infimal subproblems and organizational subunits in the real world.

2. The subproblems in the multilevel representation do correspond to organizational subunits, but this correspondence is not used in the actual solution process. For instance, in the study of distribution system design in section 8.2, there are infimal subproblems pertaining to different products. Also, the supremal subproblem may be thought of as pertaining to a special project group charged with locating warehouses. However, there is no suggestion that the product managers and the special project group should actually exchange messages with each other during the adjustment phase. Rather, a solution to the overall problem is obtained through a two-level method (the Benders decomposition method), by one set of researchers, in one location, on one computer. Messages are exchanged, not between different organizational subunits, but between different parts of one computer program.

3. There is a correspondence between organizational subunits and subproblems in the multilevel model, and this correspondence is utilized in the solution process. That is, messages are exchanged between subproblems in the adjustment phase, and the subproblems are solved and resolved by separate organizational subunits in separate locations. In this volume, several studies of this kind are discussed. In Chapter 6, two-level methods for deciding on production and sales in corporations are dealt with, and it is explicitly assumed that these methods involve the participation of different subunits in the organization (headquarters and departments), corresponding to subproblems in the two-level hierarchy. Messages are actually to be exchanged between different departments and headquarters during the adjustment phase. Similarly, in Chapter 10 a problem situation regarding the establishment of pollution levels for a set of polluters along a river is discussed. A two-level subproblem hierarchy is constructed, where the infimal subproblems pertain to polluters and the supremal subproblem pertains to a central authority in charge

of water quality control. It is assumed that the overall pollution control problem is to be solved through methods that involve the iterative physical transmittal of information between the central authority and the polluters. For yet another illustration of this sort of institutional two-level process, we may reconsider Lange's description of price determination and production planning in a socialized economy (Lange, 1939) (see section 1.3). Evidently Lange envisioned an institutional two-level process, carried out by the Central Planning Board and plant managers.

In the literature, the distinction between categories 2 and 3 above is not always made clear. This is unfortunate, since a certain confusion may result. Among sociologists and organization theorists there has been a fair amount of interest in hierarchical structures and phenomena, and it is sometimes suggested that multilevel systems analysis could be relevant to empirical studies of decision processes and organizational design (see, for instance, Baumgartner *et al.* 1975). It is clear that it is only category 3 above that could be of such relevance. In category 2, one is really not concerned with organizational phenomena at all, only with a methodology for modeling and solving decision problems, and attempts to deduce implications from that methodology for organizational design, for example, would probably not be justified.

In a category 3 situation—i.e., where the multilevel subproblem hierarchy corresponds to the hierarchy of organizational subunits and where this correspondence is actually used in the solution process—one could imagine that some of the subunits have goals of their own, conflicting with the goals of the total organization. Such subunits might then try to "cheat" (e.g., submit biased or false information in the adjustment phase) to obtain a better final result for themselves, at the expense of the organization as a whole. This means that it might be fruitful to consider game-theoretic approaches to multilevel decision making. That has, in fact, been one research direction (see, for instance, Burkov and Opoitsev 1974). However, we will not consider game-theoretic approaches in this volume. Similarly, if the subunits have private interests, one could consider the overall decision situation as one involving multiple-criterion decision making (for example, one criterion for each subunit). This means that a multicriterion approach to multilevel systems analysis might also be worthwhile. Again, such approaches are not considered in this volume. Game theory and multicriterion decision making are interesting topics in their own right, but to discuss them here would lead us away from the main line of our argument.

2.2.3 RELATED CONCEPTS

In concluding this chapter on basic ideas in multilevel systems analysis, we will mention three further concepts: decentralization, aggregation, and decomposition.

Decentralization is an important topic among organization theorists; it refers to the level in the organizational hierarchy at which decisions are made (see Jennergren 1975 for an extensive discussion of decentralization in organizations). Now suppose a multilevel method is used in a decision situation of category 3 as described in the preceding subsection. That is, the solution process is carried out through an iterative interplay between different organizational subunits. If, in the execution phase, the solution to the overall problem is derived from the supremal subproblem only, meaning that the supremal organizational subunit makes the final decision, then that is a case of centralized decision making. If, on the other hand, the solution to the overall problem is derived in the execution phase from the infimal subproblems alone, meaning that each infimal organizational subunit makes part of the decision, then one has more decentralized decision making. Decentralization is hence related to who makes the final decision in the execution phase in a situation where a multilevel method is used as an institutional decision-making process, involving different subunits in an organization.

Aggregation is related to multilevel systems analysis in two ways. First, aggregation is sometimes mentioned as an alternative to some multilevel method (Aoki 1971, p. 191; Mesarovic *et al.* 1970, p. 62; Rogers 1976). That is, if a given decision problem is very large, it may be impossible to solve it directly, in a one-level fashion. In such a situation, a multilevel solution method may be attempted, but another possible solution strategy would be to aggregate the variables and restrictions in the overall problem to obtain another problem of smaller size and then solve that problem directly, by a one-level method. Second, aggregation often occurs in multilevel systems analysis itself, in that the variables in the supremal subproblem represent a higher degree of aggregation than the variables in the infimal subproblems. For instance, in the column generation method of solving the maximal multicommodity network flow problem, the supremal subproblem operates on *paths*, whereas the infimal subproblems operate on *arcs* (to create paths for the supremal subproblem). For another example, hierarchical production scheduling (discussed in Chapter 7) is really nothing other than a hierarchical disaggregation scheme, with three levels involving successively disaggregated decision variables.

The word decomposition has different meanings. In the first place, it may denote the process of partitioning an overall decision problem $\mathscr{D}$ into subproblems $\mathscr{D}_0$ and $\mathscr{D}_1(\gamma) \ldots \mathscr{D}_n(\gamma)$. In the second place, decomposition methods are a group of mathematical programming techniques. Many of these (but not all) may be regarded as multilevel methods for modeling and solving decision problems. That is, they specify how a hierarchy of subproblems (usually with two levels) is to be constructed and how the adjustment and execution phases are to be carried out, often in an algorithmic fashion. Examples of decomposition methods that may be considered two-level procedures for modeling and solving decision problems are the Dantzig–Wolfe method, the Benders

method, and Lagrangean (price-directive) decomposition. These methods are, indeed, used in some of the case studies in this volume. They are described in the next chapter.

Decomposition methods can often be implemented as algorithmic multilevel schemes. In addition to those methods, there are other multilevel methods that must be labeled as heuristic, meaning that there may be no theoretical justification for them in terms of convergence properties, that there is no guarantee of an optimal or even a "good" solution, or that there are very few iterations of information exchange and subproblem solution in the adjustment phase. Examples of such methods are given in later chapters, particularly in Chapter 5. As an extreme case, one may consider as multilevel but heuristic methods in which there is no iterative information exchange at all. This would, in effect, imply that the adjustment phase is eliminated and that one proceeds immediately to the execution phase. As indicated in section 1.4, we have included only one study of this type in this volume (in Chapter 7 on operations management). The remaining studies incorporate an adjustment phase with at least one iteration of information exchange.

REFERENCES

Aoki, M. 1971. Aggregation, pp. 191–232. In D. A. Wismer (ed.), Optimization Methods for Large-Scale Systems. New York: McGraw-Hill.

Baumgartner, T., T. R. Burns, P. DeVille, and L. D. Meeker. 1975. A systems model of conflict and change in planning systems with multi-level, multiple objective evaluation and decision making. Yearbook of General Systems 20: 167–183.

Baumol, W. J., and T. Fabian. 1964. Decomposition, pricing for decentralization, and external economies. Management Science 11: 1–32.

Burkov, V. N., and V. I. Opoitsev. 1974. A meta-game approach to the control of hierarchical systems. (In Russian.) Avtomatika i telemekhanika No. 1: 103–114.

Dirickx, Y. M. I., L. P. Jennergren, and D. W. Peterson. 1973. Some relationships between hierarchical systems theory and certain optimization problems. IEEE Transactions on Systems, Man, and Cybernetics 3: 514–518.

Hirshleifer, J. 1956. On the economics of transfer pricing. Journal of Business 29: 172–184.

Jennergren, L. P. 1973. A price schedules decomposition algorithm for linear programming problems. Econometrica 41: 965–980.

Jennergren, L. P. 1974. On the concept of coordinability in hierarchical systems theory. International Journal of Systems Science 5: 493–497.

Jennergren, L. P. 1975. Decentralization in Organizations. Social Science Report Series No. 14. Odense, Denmark: Odense University. [To be published in P. Nystrom and W. H. Starbuck (ed.), Handbook of Organizational Design. London: Oxford University Press.]

Jennergren, L. P. 1976. The multilevel approach: A systems analysis methodology. Journal of Systems Engineering 4: 97–106.

Lange, O. 1939. On the economic theory of socialism, pp. 55–143. In B. E. Lippincott (ed.), On the Economic Theory of Socialism. Minneapolis: University of Minnesota Press.

Mesarovic, M. D., D. Macko, and Y. Takahara. 1970. Theory of Hierarchical, Multilevel Systems. New York: Academic.

Rogers, A. 1976. Shrinking large-scale population-projection models by aggregation and decomposition. Environment and Planning A 8: 515–541.

Sweeney, D. J., E. P. Winkofsky, P. Roy, and N. R. Baker. 1978. Composition vs. decomposition: Two approaches to modeling organizational decision processes. Management Science 24: 1491–1499.

3 Multilevel Solution Methods

3.1 INTRODUCTION AND OVERVIEW

As the title of this chapter indicates, we are now concerned with solution methods with multilevel features. In view of the topics dealt with in later chapters, we will be concerned with such methods insofar as they are relevant for solving mathematical programming problems. This implies that we are proposing the use of such methods for solving large-scale mathematical programming problems; however, we do not mean to create the impression that multilevel methods are the only, or even the most efficient, way of solving large mathematical programming problems. In fact, there are many algorithms with no essential multilevel aspects that have computational efficiency. In particular, such algorithms exist for solving large LP problems. We refer the reader to Lasdon (1970, Chapters 5 and 6), and Balinski and Hellerman (1975) for discussions of partitioning and compact inverse methods for linear programs. Compact inverse methods are also discussed briefly in the next chapter (section 4.1).

The methods to be reviewed in this chapter were selected only because they are used in the case studies reported in later chapters. The next section is devoted to column generation methods. It includes a discussion of the Ford–Fulkerson method for finding a maximal flow in a multicommodity network. In section 3.3 we present the Dantzig–Wolfe decomposition method.* This method is, in fact, a column generation scheme, and it is one of the most important multilevel techniques for linear problems.† It may, however, be generalized to certain nonlinear problems as well, as described in section 3.4.

* We hesitate to say "the Dantzig–Wolfe algorithm," since it is really more of a principle, or methodology, than an algorithm. Nonetheless, we will sometimes use the expression.
† The computational efficiency of the Dantzig–Wolfe method is discussed in Chapter 4.

In section 3.5, the Benders decomposition algorithm is presented. This algorithm, originally designed for mixed-integer programming problems, is adapted for linear programs with block-angular structure. The Kornai–Liptak algorithm—a simplification of the Benders algorithm—is discussed in section 3.6. The chapter continues with a presentation of the Lagrangean decomposition technique for nonlinear programming problems (section 3.7). Heuristic multilevel methods are briefly mentioned in section 3.8.

Section 3.9 discusses multilevel methods for control-theoretic problems. As indicated in section 1.4, such problems fall outside the scope of this volume, but for completeness we include a brief discussion here.

It is evident that the methods to be presented here belong to the area of decomposition in mathematical programming. For general surveys of this area, the reader is referred to Bensoussan *et al.* (1972), Geoffrion (1970a, b), Hagelschuer (1971), Lasdon (1970), and Verina and Tanaev (1975).

3.2 COLUMN GENERATION

3.2.1 GENERAL DISCUSSION

Column generation is a two-level method for solving LP problems with many columns. Suppose one is interested in solving an LP problem written in the following form

$$\begin{aligned} &\text{Maximize} && cx \\ &\text{s.t.:} && Ax \leq b, \\ & && x \geq 0, \end{aligned} \tag{3.1}$$

where A is an $m \times n$ matrix, c is a row vector of dimension n, and x and b are column vectors of dimension n and m, respectively. In the terminology of Chapter 2, (3.1) is the overall problem under consideration.

Furthermore, suppose that one has at hand a basic feasible solution to (3.1). Let π be a multiplier vector associated with the current basis. In looking for a possible better solution, one would inspect the quantity $c_r - \pi A_r$ (where c_r is the rth element of c and A_r the rth column of A), for all indices r pertaining to nonbasic variables. If $c_r - \pi A_r > 0$ for some such r, the current basis is not an optimal one. The simplex method of linear programming would at this point introduce into the basis a column pertaining to an index r such that $c_r - \pi A_r > 0$. Usually, an index r is picked for which $c_r - \pi A_r$ is maximal.

However, the linear programming problem (3.1) may have many columns—that is, the number n could be very large. In that case, it may be inefficient to store, update, and inspect information pertaining to every single column to determine which one (if any) should enter the next basis. In fact, even

attempting to write down or otherwise generate the problem (3.1) in its entirety could be a nearly impossible task. Instead, the columns should be generated as they are needed, in the algorithmic process. This is exactly what column generation methods aim at.

More precisely, suppose the number n in the specification of the problem (3.1) is very large, so that a column generation method is used. At any given iteration of the simplex algorithm, then, the complete description of problem (3.1) is not available. Instead, one has at hand the following problem specification:

$$\begin{aligned} &\text{Maximize} && c'x' \\ &\text{s.t.:} && A'x' \leq b, \\ & && x' \geq 0. \end{aligned} \tag{3.2}$$

Here, A' is an $m \times n'$ matrix composed of a subset of the columns of A. The n' vectors c' and x' are chosen correspondingly. Suppose now that a new column A_r is generated through a column generation method. Let c_r be the associated element of the c-vector. At the next iteration of the simplex algorithm, one would have the following problem specification:

$$\begin{aligned} &\text{Maximize} && (c', c_r)\begin{bmatrix} x' \\ x_r \end{bmatrix} \\ &\text{s.t.:} && (A', A_r)\begin{bmatrix} x' \\ x_r \end{bmatrix} \leq b, \\ & && \begin{bmatrix} x' \\ x_r \end{bmatrix} \geq 0. \end{aligned} \tag{3.3}$$

This extended problem can be easily solved, since by retaining the optimal solution to (3.2) and by setting $x_r = 0$, a feasible solution to (3.3) is obtained, and the simplex method can be continued with the column A_r entering in the basis.

In a two-level representation of the original problem (3.1), the supremal subproblem will be of the type (3.2): that is, it will contain only a subset of the columns of A. Depending on the problem situation, there may be one or several infimal subproblems, which generate additional columns for the supremal subproblem. The adjustment phase of column generation may be pictured as in Figure 3.1. In the execution phase, an optimal solution is obtained from the supremal subproblem.

Often, it is more natural to have several infimal subproblems, rather than just one, as will be seen in the example in the next subsection, on the multicommodity network flow problem, where a two-level structure with several subproblems arises very naturally. It is also seen in the discussion of a lot size

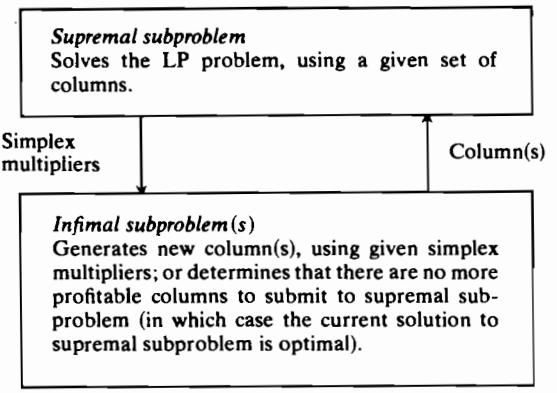


FIGURE 3.1 The adjustment phase of column generation.

production scheduling problem in Chapter 7 and a ship scheduling problem in Chapter 9. The information going down to the subproblem(s) is the vector of simplex multipliers, π, associated with the current solution to the supremal subproblem. The infimal subproblem takes the simplex multipliers and searches for an index r such that $c_r - \pi A_r$ is maximized. The column associated with that index r is the candidate for transmittal back to the supremal subproblem. If, for all infimal subproblems, there is no r such that $c_r - \pi A_r > 0$, then the current solution to the supremal subproblem is obviously optimal, in which case the solution procedure stops. We will say that one *iteration* consists of the following steps: optimization of the supremal subproblem; transmittal of a multiplier vector to infimal subproblem(s); solution of all infimal subproblems; transmittal of new columns back to the supremal subproblem. That is, one iteration always starts with optimization of the supremal subproblem.

The methods used to solve the infimal subproblem—i.e., to find that index r for which $c_r - \pi A_r$ is maximized—vary from case to case, depending on the structure of the overall problem. Often dynamic programming of one kind or another is used—for instance, a shortest-path method in the example given in the next subsection. Usually, the generated columns have natural interpretations, which arise from the overall problem situation. That is, a column could represent, for instance, one particular physical production schedule, as will be the case in the production planning model of Chapter 7.

In order to begin the solution procedure, i.e., the very first optimization of the supremal subproblem, there must, of course, be some initial columns available, sufficient in number to generate a starting basic feasible solution, with an associated multiplier vector. However, the set of initial columns could be a very small subset of the total set of all possible ones.

Column generation has been discussed here in connection with maximization problems. However, it is obviously equally applicable to overall problems

of the minimizing type. (In that case, the infimal subproblem consists in finding an index r such that $c_r - \pi A_r$ is minimal.)

3.2.2 THE MAXIMAL MULTICOMMODITY NETWORK FLOW PROBLEM

In this section we discuss the solution method originally developed by Ford and Fulkerson (1958) for finding a maximal flow in a multicommodity network. The reader will find a description of a closely related algorithm in Chapter 8, where a minimal-cost multicommodity network problem will be discussed; the ship scheduling problem in Chapter 9 utilizes yet another closely related algorithm.

Consider the network displayed in Figure 3.2. Note that we consider an undirected network, through which we assume that two commodities are to be shipped. Each commodity has its own sets of sources and sinks. In the example we have $S_1 = \{1, 2\}$ and $S_2 = \{4\}$ as the set of sources for commodity 1 and commodity 2; similarly, $T_1 = \{3\}$ and $T_2 = \{1\}$ identify the sinks. The capacities of the arcs are indicated in Figure 3.2. The objective is to maximize the sum of the flows from sources to sinks for all commodities. The assumption that the commodities are equally valued is inessential but is made for simplicity of exposition.

We may remark at this point that for the case of finding a maximal flow in a single-commodity network, very simple algorithms have been developed. The original contribution is by Ford and Fulkerson (1956). Extensive discussions of this single-level method may be found in Dantzig (1963, Chapters 19 and 20), as well as in Ford and Fulkerson (1962). The Ford–Fulkerson algorithm for single-commodity networks is not a variant of the simplex algorithm, but is a "labeling method," i.e., a method that iteratively increases the flow by giving "labels" to nodes to indicate in which direction the flow through an arc may be increased and by finding "flow-augmenting paths."

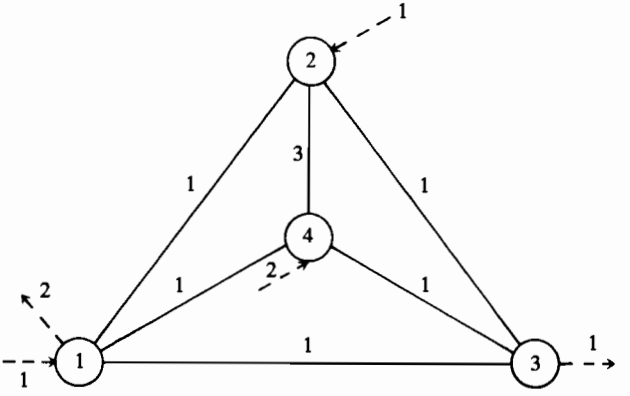


FIGURE 3.2 A multicommodity network.

The multicommodity case, however, is far more difficult.* Two approaches have been presented: single-level methods based on the "node–arc" formulation and two-level methods based on the "arc–chain" formulation. The node–arc formulation is based on stating the problem by means of conservation-of-flow equations for *each node* and capacity constraints on the flows of the various commodities passing through the arcs. Maier (1974) gives a compact inverse method based on a node–arc formulation.†

The arc–chain formulation, leading to an alternative LP formulation, will be explained here since it is the basis of the column generation scheme proposed by Ford and Fulkerson.

A *chain* is a path (a sequence of arcs) starting from a particular source of a given commodity to a particular sink of that commodity with no intermediate node appearing more than once (this is sometimes defined as an elementary chain). One can write a chain simply as a sequence of nodes with the understanding that the consecutive arcs defined by that sequence form a path. Referring to Figure 3.2, one can identify the following chains for commodity 1: {1, 3}, {2, 3}, {1, 2, 3}, {1, 4, 3}, {2, 4, 3}, {2, 1, 3}, {1, 4, 2, 3}, {1, 2, 4, 3}, {2, 4, 1, 3}, {2, 1, 4, 3}; and for commodity 2: {4, 1}, {4, 3, 1}, {4, 2, 1}, {4, 3, 2, 1}, and {4, 2, 3, 1}.

If we now consider one unit of commodity 1 shipped on the chain {2, 4, 3}, then one unit of the arc capacity of arcs (2, 4) and (3, 4) is utilized and none of the other arc capacities in the network. Thus we may associate with {2, 4, 3} a column vector, representing the capacity usage, in the following manner:

arc:		
	(1, 2)	0
	(2, 3)	0
	(1, 3)	0
	(1, 4)	0
	(2, 4)	1
	(3, 4)	1

By forming a column vector, one for each chain and each commodity, one obtains the following matrix:

0	0	1	0	0	1	0	1	0	1	0	0	1	1	0
0	1	1	0	0	0	1	0	0	0	0	0	0	1	1
1	0	0	0	0	1	0	0	1	0	0	1	0	0	1
0	0	0	1	0	0	1	0	1	1	1	0	0	0	0
0	0	0	0	1	0	1	1	1	0	0	0	1	0	1
0	0	0	1	1	0	0	1	0	1	0	1	0	1	0
					commodity 1							commodity 2		

* Mathematically, these difficulties have to do with the fact that the problem can no longer, at least in general, be formulated as an LP problem with a unimodular constraint matrix.

† The relationship between the two formulations is the following: Solving the node–arc formulation by Dantzig–Wolfe decomposition is equivalent to solving the arc–chain formulation by column generation (see Jarvis 1969).

This matrix, the arc–chain incidence matrix, leads immediately to the arc–chain LP formulation of the multicommodity flow problem.

Define variables as follows: Let x_{ij} be the amount shipped of the jth commodity on the ith chain. Obviously, $j = 1$ or 2, and $i = 1 \ldots 10$ for $j = 1$, and $i = 1 \ldots 5$ for $j = 2$. The total amount of commodity 1 shipped is $\sum_{i=1}^{10} x_{i1}$, and of commodity 2 the total amount is $\sum_{i=1}^{5} x_{i2}$. The objective is to maximize $\sum_{i=1}^{10} x_{i1} + \sum_{i=1}^{5} x_{i2}$. The resulting LP problem may be written in detached coefficient form as follows:

x_{11}	x_{21}	x_{31}	x_{41}	x_{51}	x_{61}	x_{71}	x_{81}	x_{91}	$x_{10,1}$	x_{12}	x_{22}	x_{32}	x_{42}	x_{52}	Relation	Constants
		1			1		1		1			1	1		$\leq$	1
	1	1				1							1	1	$\leq$	1
1					1			1			1			1	$\leq$	1
			1			1		1	1	1					$\leq$	1
				1		1	1	1				1		1	$\leq$	3
			1	1			1		1		1		1		$\leq$	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	max!	

Additionally, it must hold that each $x_{ij} \geq 0$. This is the overall problem. It has the optimal solution $x_{11} = x_{21} = x_{51} = x_{12} = x_{32} = 1$, all the other variables zero.

Needless to say, the number of columns to be included in the above LP problem can be enormous for networks of realistic size. Hence the arc–chain formulation is not very appealing when one wants to solve the problem in a single-level fashion. In that case, the node-arc formulation may be preferable from a computational point of view. The arc–chain formulation finds its usefulness in conjunction with column generation.

3.2.3 SOLUTION BY COLUMN GENERATION

In the maximal multicommodity network flow problem, it is most natural to associate one infimal subproblem with each commodity. The adjustment phase of the two-level solution method may be pictured as in Figure 3.3.

From the supremal subproblem, one obtains a multiplier vector with one element pertaining to each arc in the network. This vector is transmitted to each infimal subproblem. Each infimal subproblem subproblem $j = 1 \ldots n$ then searches for a column index r, or chain in the network, such that $c_{rj} - \pi A_{rj}$ is maximal, where $c_{rj} = 1$ for all j and r, π denotes the multiplier vector, and A_{rj} is a column in the constraint coefficient matrix with elements being either 0 or 1, depending on which particular arcs on the network belong to the chain. Maximizing $c_{rj} - \pi A_{rj}$ over all indices r is obviously equivalent to minimizing

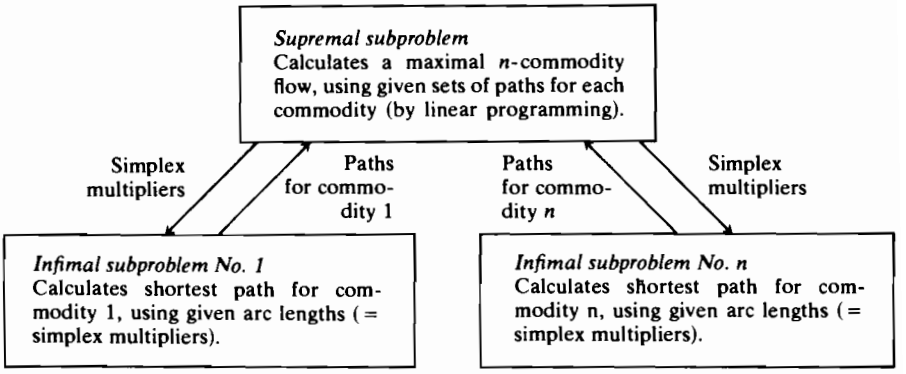


FIGURE 3.3 The adjustment phase of column generation, maximal multicommodity network flow problem.

πA_{rj}. One may now interpret the vector π as a vector of arc lengths, and minimizing πA_{rj} is hence the well-known problem of finding a shortest parth in the network. In other words, infimal subproblem j is one of finding the shortest chain from the set of sources to the set of sinks for commodity j. If that chain (which is still indexed by r) is shorter than 1, implying $c_{rj} - \pi A_{rj} > 0$, then it is transmitted to the supremal subproblem; otherwise, it is not. If, on some iteration, no infimal subproblem transmits a new chain to the supremal subproblem, the procedure stops, since an optimal solution to the overall problem is already in hand, given by the last solution to the supremal subproblem. However, if one or several of the infimal subproblems do transmit new chains to the supremal subproblem, then one or several new columns are added to that subproblem. In that case, the supremal subproblem is reoptimized, taking into account the new columns.

The infimal subproblems hence consist in finding a shortest path in the original network, where the arc lengths differ from one iteration to another. Finding a shortest path in a network is a relatively simple optimization task, for which several solution methods are available. See, for instance, Elmaghraby (1970) for a survey of some of these methods.

Consider now the example problem of the previous subsection. Since two commodities are to be shipped, there will be two infimal subproblems. To start off the algorithm, some initial columns must be available to optimize the supremal subproblem. Actually, one could begin with the slack columns, implying zero flows for both commodities as an initial basic feasible solution. Suppose, however, that the columns corresponding to the variables x_{11}, x_{21}, and x_{12} in the detached coefficient tableau in section 3.2.2 are given. The solution procedure would then be as follows.

Iteration 1. The supremal subproblem is optimized. In detached coefficient form, it may be written as follows:

x_{11}	x_{21}	x_{12}	Relation	Constants
			$\leq$	1
	1		$\leq$	1
1			$\leq$	1
		1	$\leq$	1
			$\leq$	3
			$\leq$	1
1	1	1	max!	

The solution is $x_{11} = x_{21} = x_{12} = 1$, with the associated multiplier vector $\pi = (0, 1, 1, 1, 0, 0)$. These multipliers are now interpreted as arc lengths and written into the network as shown in Figure 3.4.

Infimal subproblem 1 then consists in finding a shortest path from the set of sources for commodity 1 (nodes 1 and 2) to the sink, node 3. One such path is $\{2, 4, 3\}$ with length 0. It may be represented as a column, for convenience written as a row vector $(0, 0, 0, 0, 1, 1)$,* which is reported back to the supremal subproblem. The path pertains to variable x_{51} in the detached coefficient tableau of the overall problem.

Infimal subproblem 2 consists in finding a shortest path from the source for commodity 2 (node 4) to the sink (node 1) (there is only one shortest path in this case). That path is found to be $\{4, 2, 1\}$, with a total length of 0. It may be written as a (transposed) column $(1, 0, 0, 0, 1, 0)$. It, too, is reported back to the

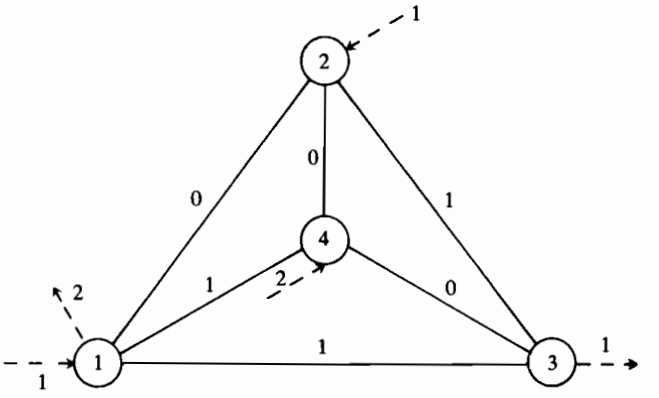


FIGURE 3.4 The network for infimal optimization.

* This convention will be used sometimes in this chapter.

supremal subproblem. It pertains to variable x_{32} in the overall problem formulation, above.

Iteration 2. The supremal subproblem now has two new columns. It may be formulated as follows, in detached coefficient form:

x_{11}	x_{21}	x_{51}	x_{12}	x_{32}	Relation	Constants
				1	$\leq$	1
	1				$\leq$	1
1					$\leq$	1
			1		$\leq$	1
		1		1	$\leq$	3
		1			$\leq$	1
1	1	1	1	1	max!	

The solution is now $x_{11} = x_{21} = x_{51} = x_{12} = x_{32} = 1$, and the associated multiplier vector $\pi = (1, 1, 1, 1, 0, 1)$. The arc lengths in the network are changed correspondingly. As in iteration 1, infimal subproblem 1 consists in finding the shortest path from node 1 or 2 to node 3, but where the arc lengths are now given by $\pi = (1, 1, 1, 1, 0, 1)$. At this point, there is no path shorter than 1, so there is no path to report back to the supremal subproblem. Infimal subproblem 2 consists in finding the shortest path from node 4 to node 1. Again, there is no path shorter than 1, so no path is reported to the supremal subproblem from this infimal subproblem either. This means that an optimal solution to the overall problem has been attained. It is given by the last solution to the supremal subproblem, $x_{11} = x_{21} = x_{51} = x_{12} = x_{32} = 1$, with all other possible variables equal to zero. This solution can easily be translated into physical flows. One unit of commodity 1 should be sent along {1, 3}; one unit of commodity 1 along {2, 3}; one unit of commodity 1 along {2, 4, 3}; one unit of commodity 2 along {4, 1}; and one unit of commodity 2 along {4, 2, 1}. We might mention here that the interpretation of an optimal solution obtained from the arc–chain formulation is easier than for the node–arc formulation, especially for large networks.

3.3 THE DANTZIG–WOLFE DECOMPOSITION METHOD FOR LINEAR PROGRAMS

The Dantzig–Wolfe method, first described in the seminal contribution of Dantzig and Wolfe (1961), is probably the most important multilevel method for solving large-scale linear programming problems. However, in terms of computational efficiency it may be challenged by the single-level, compact

inverse methods that originated with the work of Dantzig and Van Slyke (1967) (see Chapter 4).

3.3.1 THE REPRESENTATION OF A POLYHEDRAL CONVEX SET*

As is well known, a polyhedral convex set X may be defined as the intersection of halfspaces, or

$$X = \{x \mid Bx \leq b\},$$

where B is a matrix and b a vector of suitable dimensions. To avoid trivialities, we assume X to be nonempty. Let $x^1 \ldots x^P$ be the set of extreme points of X and let $\tilde{x}^1 \ldots \tilde{x}^R$ identify the extreme rays of the polyhedral convex cone $\{x \mid Bx \leq 0\}$ (the R extreme rays are defined by the halflines $\delta \tilde{x}^r$, $\delta \geq 0$). It can be demonstrated that

$$X = \left\{ x \,\middle|\, x = \sum_{p=1}^{P} \lambda^p x^p + \sum_{r=1}^{R} \delta^r \tilde{x}^r, \text{ with } \sum_{p=1}^{P} \lambda^p = 1, \lambda^p \geq 0, \delta^r \geq 0, p = 1 \ldots P, r = 1 \ldots R \right\}.$$

(If R is zero, the corresponding summation vanishes. It is assumed that $P \neq 0$.) This result states that the set X may be regarded as the sum of a bounded polyhedral convex set and a polyhedral convex cone. If the cone is empty, i.e., if there is no x satisfying the inequalities $Bx \leq 0$, then X is bounded.

To illustrate the above characterization, consider the set

$$X = \{(x_1, x_2) \mid -2x_1 + x_2 \leq 1,\ x_1 - x_2 \leq 1,\ x_1 \geq 0,\ x_2 \geq 0\},$$

which is graphed in Figure 3.5. This set has three extreme points: (0, 0), (0, 1), and (1, 0). The associated convex cone has two extreme rays passing through the points (1, 2) and (1, 1). The resulting bounded polyhedral convex set and the convex cone are displayed in Figure 3.5.

3.3.2 AN OUTLINE OF THE DANTZIG-WOLFE DECOMPOSITION METHOD

Now consider some arbitrary linear program, written as

$$\begin{aligned} &\text{Minimize} && cx \\ &\text{s.t.:} && Ax = b, \\ & && x \geq 0. \end{aligned} \tag{3.4}$$

* The characterization of a polyhedral convex set is central to the Dantzig–Wolfe method. The reader who finds the present discussion too sketchy or too advanced may benefit from the excellent discussion in Appendix B of Simonnard (1966).

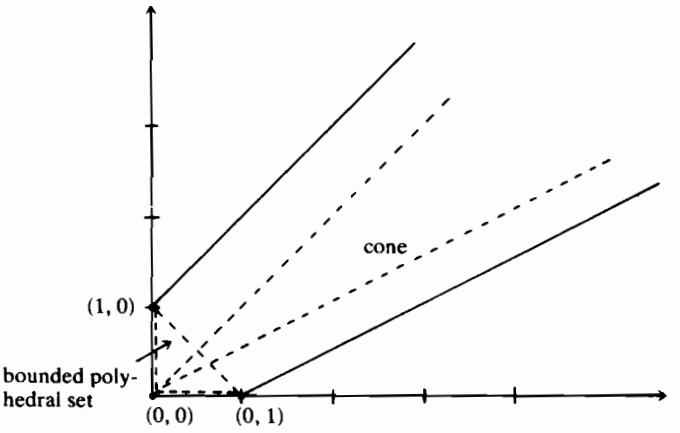


FIGURE 3.5 The representation of a polyhedral convex set.

Problem (3.4) is the overall problem under consideration. Suppose A has m rows and that this matrix is partitioned as

$$\begin{bmatrix} A_1 \\ A_2 \end{bmatrix},$$

where A_1 has m_1 rows and A_2 has m_2 rows ($m_1 + m_2 = m$). The vector b is similarly partitioned. One may then rewrite (3.4) as

$$\begin{aligned} &\text{Minimize} && cx \\ &\text{s.t.:} && A_1 x = b_1, \\ &&& A_2 x = b_2, \\ &&& x \geq 0. \end{aligned} \tag{3.5}$$

The set $X_2 = \{x \mid A_2 x = b_2,\ x \geq 0\}$ is a polyhedral convex set, just like X of the previous subsection. Let $x^1 \ldots x^P$ be the set of its extreme points and $\tilde{x}^1 \ldots \tilde{x}^R$ identify the set of extreme rays of the associated convex cone. This means that any $x \in X_2$ can be written as

$$x = \sum_{p=1}^{P} \lambda^p x^p + \sum_{r=1}^{R} \delta^r \tilde{x}^r, \tag{3.6}$$

with $\sum \lambda^p = 1$, $\lambda^p \geq 0$, $\delta^r \geq 0$. For any such choices of the δ- and λ-variables, the resulting $x \in X_2$. Using (3.6), if we substitute for x in the objective function cx

and the constraint $A_1x = b_1$ of (3.5), we obtain the following equivalent problem:

$$\begin{aligned} \text{Minimize} \quad & c\Big(\sum_{p=1}^{P} \lambda^p x^p + \sum_{r=1}^{R} \delta^r \tilde{x}^r\Big) \\ \text{s.t.:} \quad & A_1\Big(\sum_{p=1}^{P} \lambda^p x^p + \sum_{r=1}^{R} \delta^r \tilde{x}^r\Big) = b_1, \\ & \sum \lambda^p = 1, \quad \lambda^p \geq 0, \quad \delta^r \geq 0. \end{aligned}$$

Making use of the convention that $w^p = cx^p$, $\tilde{w}^r = c\tilde{x}^r$, $L^p = A_1x^p$, and $\tilde{L}^r = A_1\tilde{x}^r$, this problem can be written as

$$\begin{aligned} \text{Minimize} \quad & \sum_{p=1}^{P} w^p\lambda^p + \sum_{r=1}^{R} \tilde{w}^r\delta^r \\ \text{s.t.:} \quad & \sum_{p=1}^{P} L^p\lambda^p + \sum_{r=1}^{R} \tilde{L}^r\delta^r = b_1, \qquad (3.7) \\ & \sum_{p=1}^{P} \lambda^p = 1, \quad \lambda^p \geq 0, \quad \delta^r \geq 0. \end{aligned}$$

Problems (3.4) and (3.7) are equivalent. They have the same optimal solution value. Any feasible solution to (3.7) corresponds to a feasible solution to (3.4), which may be recovered by using (3.6). Conversely, given a feasible solution $\bar{x}$ to (3.4), there exists at least one feasible solution $\bar{\lambda}^p$, $\bar{\delta}^r$ to (3.7) such that $\bar{x} = \sum \bar{\lambda}^p x^p + \sum \bar{\delta}^r \tilde{x}^r$.

Let us pause to see what has been accomplished. The problem (3.4), which has m rows, has been replaced by (3.7), which has only $m_1 + 1$ rows. However, the problem (3.7) has as many variables as there are extreme points and extreme rays in the set X_2. This could be a very large number indeed—much larger than the number of variables in (3.4).

The Dantzig–Wolfe decomposition method solves (3.7) rather than (3.4), but since (3.7) could have a great many variables, it is not desirable to write down the complete specification of (3.7) in advance. Rather, the columns of (3.7) should be generated as they are needed. We now turn to the mechanism for generating these columns.

Suppose that at some stage in the process of solving (3.7), one has generated a subset of all the columns of that problem. Those problems correspond to certain extreme points and extreme rays of the set X_2. Suppose $\bar{p}$ columns corresponding to extreme points and $\bar{r}$ columns corresponding to extreme rays have been generated, where $\bar{p} + \bar{r} < P + R$. One then has at hand the following

problem specification:

$$\begin{aligned} \text{Minimize} \quad & \sum_{p=1}^{\bar{p}} w^p \lambda^p + \sum_{r=1}^{\bar{r}} \tilde{w}^r \delta^r \\ \text{s.t.:} \quad & \sum_{p=1}^{\bar{p}} L^p \lambda^p + \sum_{r=1}^{\bar{r}} \tilde{L}^r \delta^r = b_1, \\ & \sum_{p=1}^{\bar{p}} \lambda^p = 1, \quad \lambda^p \geq 0, \quad \delta^r \geq 0. \end{aligned} \tag{3.8}$$

Problem (3.8) is now optimized. Now a question arises: Is the optimal solution to (3.8) also an optimal solution to (3.7)? Let π be an m_1-vector of optimal dual multipliers pertaining to the first m_1 equality restrictions of (3.8), and α an optimal dual multiplier pertaining to the last equality constraint. The usual simplex method optimality criterion is applied—that is, one wants to determine whether there exists some extreme point $x^{p'}$ of X_2 such that $w^{p'} - \pi L^{p'} - \alpha < 0$, or some extreme ray $\tilde{x}^{r'}$ such that $\tilde{w}^{r'} - \pi \tilde{L}^{r'} < 0$. If the answer to both questions is no, the optimality test has been passed, and the optimal solution to (3.8) is indeed optimal for (3.7) as well. This also means that one may recover an optimal solution to the original problem (3.4) by means of the relation (3.6).

However, if there does exist $x^{p'}$ such that $w^{p'} - \pi L^{p'} < \alpha$ or $\tilde{x}^{r'}$ such that $\tilde{w}^{r'} - \pi L^{r'} < 0$, then we cannot conclude that the current optimal solution to (3.8) is also optimal for (3.7), and the solution process must continue. The simplex method would now look for that variable that has the smallest relative cost factor, i.e., that index p' or r' for which $(w^{p'} - \pi L^{p'} - \alpha)$ or $(\tilde{w}^{r'} - \pi \tilde{L}^{r'})$ is minimal. If the minimum is attained for p' (corresponding to an extreme point), then the column $(L^{p'}, 1)$ with associated objective function coefficient $w^{p'}$ should be entered into the basis of the linear program (3.8) at the next iteration. If the minimum is attained for r' (corresponding to an extreme ray), the column $(\tilde{L}^{r'}, 0)$ with associated objective function coefficient $\tilde{w}^{r'}$ is introduced into the basis.

In any case, for the optimality test as well as for identifying a new column to introduce into the basis, one must find that index p' or r' for which $(w^{p'} - \pi L^{p'} - \alpha)$ or $(\tilde{w}^{r'} - \pi \tilde{L}^{r'})$ is minimal. But this can obviously be done by considering the following linear program:

$$\begin{aligned} \text{Minimize} \quad & (c - \pi A_1)x \\ \text{s.t.:} \quad & A_2 x = b_2, \\ & x \geq 0. \end{aligned} \tag{3.9}$$

If this problem has a finite optimal solution, then the optimum is taken on at an extreme point of the set X_2, and that extreme point will be found when (3.9) is solved by the simplex method. If (3.9) has unbounded solutions, then the

optimum is taken on along an extreme ray of the set X_2. That extreme ray, too, is identified when (3.9) is solved by the ordinary simplex method. (This matter will be discussed in somewhat greater detail in the next section.)

The Dantzig–Wolfe decomposition method, as applied to problem (3.4), may be outlined step by step:

Step 1 (start of a new iteration). Solve problem (3.8), using only those columns which are at hand at this point. Let π and α be optimal dual variables. [If (3.8) has an unbounded optimal solution, then so has the original problem (3.4). In that case, stop.]

Step 2. Solve problem (3.9). If (3.9) has an unbounded solution along the extreme ray identified by $\tilde{x}^{r'}$, go to step 3. If (3.9) has a finite optimal solution, the extreme point $x^{p'}$, go to Step 4. [If (3.9) has no feasible solution at all, then stop—the original problem (3.4) also has no feasible solution.]

Step 3. Add the column $(\tilde{L}^{r'}, 0)$ with associated objective function coefficient $\tilde{w}^{r'}$ to problem (3.8). Return to Step 1.

Step 4. Carry out the simplex method optimality test: If $w^{p'} - \pi L^{p'} \geq \alpha$, the current solution to (3.8) is also optimal for (3.7). In that case, stop—an optimal solution to (3.4) may be recovered through (3.6). If $w^{p'} - \pi L^{p'} < \alpha$, the current solution to (3.8) is not optimal for (3.7) (except under certain degeneracy conditions). Then add the column $(L^{p'}, 1)$ to problem (3.8). The associated objective function coefficient is $w^{p'}$. Go back to Step 1.

In conclusion, it may be remarked that problem (3.7) is often referred to as the *extremal problem*, or the *full master problem*, equivalent to the original problem. Problem (3.8), which includes only a subset of the columns of (3.7), is often called the *restricted master problem*. In the terminology of Chapter 2, the restricted master problem is also the supremal subproblem, and (3.9) is the infimal subproblem. In the adjustment phase, simplex multipliers are transmitted from the supremal subproblem to the infimal one, and columns for the supremal subproblem are sent back from the infimal subproblem. The adjustment phase of the Dantzig–Wolfe method could hence be visualized as shown in Figure 3.1. In the execution phase, a solution to the overall problem (3.4) is recovered from the supremal subproblem by means of (3.6). As was pointed out in the discussion of column generation in section 3.2.1, an iteration starts with the optimization of the supremal subproblem.

3.3.3 LINEAR PROGRAMMING PROBLEMS WITH UNBOUNDED SOLUTIONS

Step 2 of the Dantzig–Wolfe decomposition method outlined in the preceding subsection requires solution of the linear program (3.9). If this problem has a finite optimal solution, then the optimum is taken on at an extreme point of X_2, and it is well known that the simplex method will identify some extreme

point as being the optimal solution. However, if the optimum is unbounded, then it is taken on along an extreme ray. That extreme ray, too, will be identified by the simplex method. Consider the following example:

$$\begin{aligned} &\text{Maximize} && x_1 + x_2 \\ &\text{s.t.:} && -2x_1 + x_2 + x_3 = 1, \\ & && x_1 - x_2 + x_4 = 1, \\ & && x_1, x_2, x_3, x_4 \geq 0. \end{aligned}$$

The first two iterations of the simplex method (using the tableau format of Dantzig 1963) are given below. The variable x_2 is pivoted into the basis as indicated by the circled entry in the first tableau.

	x_1	x_2	x_3	x_4	$-z$	Constants
x_3	-2	①	1	0	0	1
x_4	1	-1	0	1	0	1
$-z$	-1	-1	0	0	1	0
x_2	-2	1	1	0	0	1
x_4	-1	0	1	1	0	2
$-z$	-3	0	1	0	1	1

One now discovers that the problem has an unbounded solution. As one tries to pivot the x_1 column into the basis, one sees that x_2 and x_4 increase in value. For $x_1 = 1$, one obtains $x_2 = 1 + 2$, and $x_4 = 2 + 1$. In general, for $x_1 = k > 0$, one obtains $x_2 = 1 + 2k$ and $x_4 = 2 + k$. From this we infer that $(x_1, x_2, x_3, x_4) = (k, 2k, 0, k)$ is a ray, and it is, in fact, an extreme ray.

To demonstrate this last statement, we utilize the following result from Gale (1960, p. 65): Let x be an n vector. The solution $\bar{x}$ to the inequality $Bx \leq 0$ is an extreme ray if and only if the set of rows b^i of B for which $b^i\bar{x} = 0$ has rank $n - 1$. In the present case, $n = 4$. The constraint set is given by the restrictions

$$\begin{aligned} -2x_1 + x_2 + x_3 &= 1, \\ x_1 - x_2 + x_4 &= 1, \\ x_1 &\geq 0, \\ x_2 &\geq 0, \\ x_3 &\geq 0, \\ x_4 &\geq 0. \end{aligned}$$

The associated cone is defined by

$$\begin{aligned} -2x_1 + x_2 + x_3 &= 0, \\ x_1 - x_2 + x_4 &= 0, \\ x_1 &\geq 0, \\ x_2 &\geq 0, \\ x_3 &\geq 0, \\ x_4 &\geq 0. \end{aligned} \tag{3.10}$$

Consider now the solution $(x_1, x_2, x_3, x_4) = (1, 2, 0, 1)$ to the set of restrictions (3.10). The rank of the coefficient matrix associated with those restrictions for which equality holds (the first two equalities and $x_3 = 0$) is $3 = 4 - 1 = n - 1$, so, using Gale's result, it follows that $(1, 2, 0, 1)$ identifies an extreme ray.

By a suitable generalization of this example, it is easy to see that if the problem under consideration has an unbounded optimal solution, then an appropriate ray may be picked out of the final simplex tableau.

3.3.4 A NUMERICAL EXAMPLE

The following example was used by Dantzig (1963, pp. 455–461), in a rather amusing passage with the title "Decomposition Principle, Animated." Let the original problem that one wants to solve be:

Minimize $3x_{11} + 6x_{21} + 6x_{31} + 5x_{41} + 8x_{12} + x_{22} + 3x_{32} + 6x_{42}$

s.t.:

$$\left.\begin{aligned} 2x_{31} + 2x_{22} &\leq 9, \\ x_{11} + x_{21} + x_{31} + x_{41} &= 9, \\ x_{12} + x_{22} + x_{32} + x_{42} &= 8, \\ x_{11} + x_{12} &= 2, \\ x_{21} + x_{22} &= 7, \\ x_{31} + x_{32} &= 3 \\ x_{41} + x_{42} &= 5 \end{aligned}\right\} \tag{3.11}$$

$$\text{all } x_{ij} \geq 0.$$

It is seen that this problem has a constraint set that naturally divides into two groups of restrictions. The first group corresponds to the restriction $A_1 x = b_1$ of (3.5) and consists of the single inequality $2x_{31} + 2x_{22} \leq 9$. The remaining restrictions, corresponding to $A_2 x = b_2$ in (3.5), have a very simple structure since they define the feasible region of a transportation problem. That is, if it were not for the single restriction $2x_{31} + 2x_{22} \leq 9$, the whole problem could be

solved directly, by the transportation method. Nevertheless, the Dantzig–Wolfe decomposition method offers one way of utilizing the special structure of the six transportation restrictions (3.11).

Suppose, then, that one wants to solve this problem by the Dantzig–Wolfe decomposition method, with the first restriction $2x_{31}+2x_{22}\leq 9$ incorporated into the extremal problem. To begin, a few columns of the extremal problem must be at hand at the outset. In this case, the set of $x_{ij}\geq 0$ satisfying the six transportation equalities is obviously bounded, so there will be no columns in the extremal problem corresponding to extreme rays. One extreme point may be obtained by simply ignoring the restriction $2x_{31}+2x_{22}\leq 9$ and then optimizing the remaining transportation problem. It is found to be $x^1=(x_{11}, x_{21}, x_{31}, x_{41}, x_{12}, x_{22}, x_{32}, x_{42})=(2, 0, 2, 5, 0, 7, 1, 0)$. Another extreme point is $x^2=(2, 7, 0, 0, 0, 0, 3, 5)$. With these two extreme points, we compute $cx^1=53$, $A_1x^1=18$; $cx^2=87$, $A_1x^2=0$. We can then write down the first restricted master problem as follows:

$$\begin{aligned} &\text{Minimize} && 53\lambda^1+87\lambda^2 \\ &\text{s.t.:} && 18\lambda^1 \qquad \leq 9, \\ & && \lambda^1+\lambda^2=1, \\ & && \lambda^1,\lambda^2\geq 0. \end{aligned}$$

Iteration 1. The first restricted master problem is solved. The optimal solution is $\lambda^1=1/2$, $\lambda^2=1/2$. The optimal dual multiplier π associated with the constraint $18\lambda^1\leq 9$ is $-17/9$, the optimal multiplier α associated with the constraint $\lambda^1+\lambda^2=1$ is 87. To test whether the current solution is also optimal for the full master problem, and if it is not, to find a new column to add to the restricted master problem, one solves the transportation problem:

$$\begin{aligned} &\text{Minimize} && 3x_{11}+6x_{21}+(6+(34/9))x_{31}+5x_{41}+8x_{12} \\ & && +(1+(34/9))x_{22}+3x_{32}+6x_{42} \\ &\text{s.t.:} && (3.11) \text{ and } x_{ij}\geq 0. \end{aligned}$$

The optimal solution here is $x^3=(2, 2, 0, 5, 0, 5, 3, 0)$, $w^3-\pi L^3=57+18\frac{8}{9}=75\frac{8}{9}<87=\alpha$, so the column $(L^3, 1)=(10, 1)$ is added to the restricted master problem. The associated objective function coefficient is $w^3=cx^3=57$.

Iteration 2. The restricted master problem is now

$$\begin{aligned} &\text{Minimize} && 53\lambda^1+87\lambda^2+57\lambda^3 \\ &\text{s.t.:} && 18\lambda^1 \qquad +10\lambda^3\leq 9, \\ & && \lambda^1+\lambda^2+\quad \lambda^3=1, \\ & && \lambda^1,\lambda^2,\lambda^3\geq 0. \end{aligned}$$

The optimal solution is $\lambda^1 = 0$, $\lambda^2 = 1/10$, $\lambda^3 = 9/10$. The optimal dual multiplier π associated with the constraint $18\lambda^1 + 10\lambda^3 \leq 9$ is -3. The optimal dual multiplier α associated with the constraint $\lambda^1 + \lambda^2 + \lambda^3 = 1$ is 87. To test whether the current solution to the restricted master problem is optimal for the full master problem, we must again solve a transportation problem:

$$\begin{aligned} \text{Minimize} \quad & 3x_{11} + 6x_{21} + (6+6)x_{31} + 5x_{41} + 8x_{12} \\ & + (1+6)x_{22} + 3x_{32} + 6x_{42} \\ \text{s.t.:} \quad & (3.11) \text{ and } x_{ij} \geq 0. \end{aligned}$$

This problem has many optimal solutions, but one optimal solution is $x^3 = (2, 2, 0, 5, 0, 5, 3, 0)$, i.e., the same optimal solution to the transportation problem as was obtained in the previous iteration. For $\pi = -3$ and $\alpha = 87$, one obviously obtains $w^3 - \pi L^3 - \alpha = 57 + 30 - 87 = 0$. This means that the optimality test has been passed, so $\lambda^1 = 0$, $\lambda^2 = 1/10$, $\lambda^3 = 9/10$ is an optimal solution to the full master problem, or the extremal problem. Consequently, an optimal solution to the original problem may be computed as: $(1/10)x^2 + (9/10)x^3 = (1/10)(2, 7, 0, 0, 0, 0, 3, 5) + (9/10)(2, 2, 0, 5, 0, 5, 3, 0) = (2, 2.5, 0, 4.5, 0, 4.5, 3, 0.5)$.

3.3.5 SOME FURTHER REMARKS ON THE DANTZIG–WOLFE DECOMPOSITION METHOD

It is evident from the previous discussion that the Dantzig–Wolfe decomposition method is founded on the following two basic ideas:

- The original problem is replaced by another equivalent one, the extremal, or full master, problem.
- The extremal problem is solved through a column-generation scheme.

The Dantzig–Wolfe decomposition method is thus a special case of column generation. In that respect, it may be considered to be inspired by the Ford–Fulkerson algorithm for multicommodity network flows, which is also an instance of column generation; this explains the historical importance of the Ford–Fulkerson paper. In column generation, various methods may be utilized for the successive selection of additional columns, as already indicated in section 3.2.1. In the Dantzig–Wolfe decomposition method, the simplex method itself is used for this purpose.

As already indicated in section 3.3.2, the Dantzig–Wolfe decomposition method may be regarded as a two-level method, just like any other column generation method. For instance, in the numerical example in the preceding subsection, one switches between a restricted master problem of ordinary LP type and a transportation problem. An obvious two-level arrangement is to

take the restricted master problem as the supremal subproblem and the transportation problem as the infimal subproblem. In the adjustment phase, the supremal subproblem successively receives extreme points (which may be interpreted as plan proposals) from the infimal subproblem. It mixes those plan proposals already at hand in an optimal fashion. As a by-product of this mixing, dual variables are obtained which are transferred to the infimal subproblem as guidance for the generation of further plan proposals. In the execution phase, the final decision on a shipping plan that is optimal for the original problem is made by means of the supremal subproblem.

In the earlier description of the Dantzig–Wolfe decomposition method, it was implicitly assumed that there is at hand at the outset a set of columns of the restricted master problem such that the problem has a feasible solution. The restricted master problem can then be optimized immediately at the beginning of the algorithm, and optimal dual multipliers can be derived. These dual multipliers are used to generate the first new column. If one does not have at hand at the start a set of columns that allow for a feasible solution to the restricted master problem, one can initiate the Dantzig–Wolfe decomposition method with an ordinary Phase I procedure (see Dantzig 1963). That is, as the starting basis of the restricted master problem one takes a set of artificial columns. It is then the object of Phase I to drive these columns out of the basis. Phase I terminates with a feasible solution to the restricted master problem or with the information that no such solution exists. In the latter case, the original problem (the one that one is trying to solve by the Dantzig–Wolfe decomposition method) also has no feasible solution. Once a feasible solution to the restricted master problem is obtained, all successive later iterations also involve feasible solutions to the restricted master problem. This means that the method may be terminated prior to reaching an optimal solution. A nonoptimal but feasible solution to the original problem (3.4) can then be reconstructed by means of the relation (3.6).

Convergence to an optimal solution is guaranteed in a finite number of iterations (assuming away pathological degeneracy cases). Indeed, since the extremal problem is an ordinary LP problem, it has only a finite number of different basic feasible solutions. However, the number of iterations needed to solve the original problem (3.4) by Dantzig–Wolfe decomposition could well be much larger than the number required by an application of the ordinary simplex method to that problem (see Adler and Ulkücü 1973 on this subject). For this reason, it may be desirable to terminate before optimality is reached. An attractive feature of the Dantzig–Wolfe decomposition method in this connection is that it provides a bound on the optimal solution value.

To explore this bound property, suppose that one is still interested in solving the problem (3.4) and that the restricted master problem at some iteration t has optimal solution value z_t. Let π and α be optimal dual multipliers associated with the restrictions of the restricted master problem in that iteration. Let $\bar{\lambda}^p$

$(p = 1 \ldots P)$ and $\bar{\delta}^r$ $(r = 1 \ldots R)$ be an optimal solution to the extremal problem. From the specification of that problem, it follows that

$$\bar{z} = \sum_{p=1}^{P} w^p \bar{\lambda}^p + \sum_{r=1}^{R} \tilde{w}^r \bar{\delta}^r, \tag{3.12}$$

$$b_1 = \sum_{p=1}^{P} L^p \bar{\lambda}^p + \sum_{r=1}^{R} \tilde{L}^r \bar{\delta}^r, \tag{3.13}$$

and

$$1 = \sum_{p=1}^{P} \bar{\lambda}^p. \tag{3.14}$$

Hence $\bar{z}$ is the optimal solution value for the extremal problem and also for problem (3.4). Now multiply (3.13) by π and (3.14) by α and subtract from (3.12), to obtain

$$\bar{z} - (\pi b_1 + \alpha) = \sum_{p=1}^{P} \bar{\lambda}^p (w^p - \pi L^p - \alpha) + \sum_{r=1}^{R} \bar{\delta}^r (\tilde{w}^r - \pi \tilde{L}^r).$$

Noting that $z_t = \pi b_1 + \alpha$, it holds that

$$\bar{z} \geq z_t + \min_{p=1 \ldots P} (w^p - \pi L^p - \alpha) + \sum_{r=1}^{R} \bar{\delta}^r [\min_{r=1 \ldots R} (\tilde{w}^r - \pi \tilde{L}^r)].$$

Now consider the infimal subproblem in the present iteration, i.e., problem (3.9), rewritten here for convenience:

$$\begin{aligned} &\text{Minimize} && (c - \pi A_1)x \\ &\text{s.t.:} && A_2 x = b_2, \\ & && x \geq 0. \end{aligned}$$

If this problem has an unbounded solution, no lower bound on $\bar{z}$ can be obtained in the present iteration. However, suppose it has a bounded solution so that either $\min_{r=1 \ldots R} (\tilde{w}^r - \pi \tilde{L}^r) \geq 0$ or $R = 0$, which allows us to write

$$\bar{z} \geq z_t + \min_{p=1 \ldots P} (w^p - \pi L^p - \alpha) \equiv \tilde{z}_t. \tag{3.15}$$

Inequality (3.15) gives us a lower bound on $\bar{z}$. The quantities $w^p - \pi L^p - \alpha$ are automatically produced in solving (3.9) so there is no extra work involved in computing z_t. Obviously, $\tilde{z}_t \leq \bar{z} \leq z_t$.

The sequence z_t converges monotonically to $\bar{z}$. The sequence $\tilde{z}_t$ also converges to $\bar{z}$, but not monotonically. At the iteration at which an optimal solution to the extremal problem is obtained, $z_t = \tilde{z}_t$. The quantities z_t and $\tilde{z}_t$ may be used to construct a simple stopping rule: terminate the algorithm if $z_t - \tilde{z}_* < \varepsilon > 0$, where ε is some tolerance constant that has been set in advance, and $\tilde{z}_*$ is

the best lower bound obtained so far. Such termination before optimality is reached may be desirable, as mentioned earlier.

As an example, we compute the lower bound $\tilde{z}_1$ in the example in section 3.3.4. The optimal solution value of the restricted master problem in the first iteration, z_1, is 70, so that (3.15) gives

$$\tilde{z}_1 = 70 + (57 - (-17/9)10 - 87) = 58\tfrac{8}{9}.$$

This is not a bad lower bound, since the optimal solution value $\bar{z}$ is 60.

3.3.6 BLOCK-ANGULAR STRUCTURES

Consider now a somewhat different problem formulation, one that has a block-angular structure:

$$\begin{aligned}
\text{Maximize} \quad & c_1x_1 + c_2x_2 + \cdots + c_nx_n \\
\text{s.t.:} \quad & A_1x_1 + A_2x_2 + \cdots + A_nx_n \leq a, \\
& B_1x_1 \qquad\qquad\qquad\quad \leq b_1, \\
& \qquad B_2x_2 \qquad\qquad\quad \leq b_2, \\
& \qquad\qquad \cdots \\
& \qquad\qquad\qquad B_nx_n \leq b_n, \\
& x_1, x_2 \ldots x_n \geq 0.
\end{aligned} \tag{3.16}$$

This is the overall problem considered in this subsection. Here, x_j is an n_j vector, A_j is an $m \times n_j$ matrix, B_j is an $m_j \times n_j$ matrix, b_j is an m_j vector for $j = 1 \ldots n$, and a is an m vector. Apparently, problem (3.16) has a rather special structure: if it were not for the coupling restrictions $A_1x_1 + A_2x_2 + \cdots + A_nx_n \leq a$, (3.16) would divide into n independent, smaller LP problems. The Dantzig–Wolfe decomposition method offers a way of exploiting this block-angular structure. Indeed, block-angular problem structures are considered particularly suited to that method.

Consider the set $X_j = \{x_j \mid B_jx_j \leq b_j,\ x_j \geq 0\}$. As before, any $x_j \in X_j$ may be expressed as

$$x_j = \sum_{p=1}^{P(j)} \lambda_j^p x_j^p + \sum_{r=1}^{R(j)} \delta_j^r \tilde{x}_j^r,$$

where x_j^p $[p = 1 \ldots P(j)]$ are the extreme points of X_j and $\tilde{x}_j^r$ $[r = 1 \ldots R(j)]$ identify the extreme rays of $\{x_j \mid B_jx_j \leq 0,\ x_j \geq 0\}$. Using a notational convention similar to that used in section 3.3.2, it is not difficult to see that (3.16) can be

reformulated as the following equivalent extremal problem:

$$\begin{aligned} \text{Maximize} \quad & \sum_{j=1}^{n} \left(\sum_{p=1}^{P(j)} w_j^p \lambda_j^p + \sum_{r=1}^{R(j)} \tilde{w}_j^r \delta_j^r \right) \\ \text{s.t.:} \quad & \sum_{j=1}^{n} \left(\sum_{p=1}^{P(j)} L_j^p \lambda_j^p + \sum_{r=1}^{R(j)} \tilde{L}_j^r \delta_j^r \right) \leq a, \\ & \sum_{p=1}^{P(j)} \lambda_j^p = 1 \qquad \text{for } j = 1 \ldots n, \\ & \text{all } \lambda_j^p \geq 0,\ \delta_j^r \geq 0. \end{aligned} \tag{3.17}$$

Problem (3.17) is very similar to the extremal problem (3.7). The main difference is that in (3.17), n sets X_j are expressed in terms of extreme points and extreme rays; in (3.7), there is only one such set. As a consequence, (3.17) has n restrictions of the type $\sum_{p=1}^{P(j)} \lambda_j^p = 1$. In (3.7), there is only one such restriction.

Again, the Dantzig–Wolfe decomposition method solves (3.17) through a column generation scheme. Suppose π is an optimal dual multiplier vector associated with the first m inequality restrictions of the restricted master program at some iteration, and let $\alpha = (\alpha_1 \ldots \alpha_n)$ be an optimal dual multiplier vector associated with the equality constraints.

For each index $j = 1 \ldots n$, the following problem is now solved to identify a possible new column for the restricted master problem:

$$\begin{aligned} \text{Maximize} \quad & (c_j - \pi A_j) x_j \\ \text{s.t.:} \quad & B_j x_j \leq b_j, \\ & x_j \geq 0. \end{aligned} \tag{3.18}$$

This problem is completely analogous to problem (3.9) above. If this problem has feasible solutions tending to $+\infty$ along some extreme ray $\tilde{x}_j^r$ of the set X_j, one adds $(\tilde{L}_j^r, 0 \ldots 0)$ as a new column to the restricted master problem. This column has $m+n$ components. The associated objective function coefficient is $\tilde{w}_j^r$. If (3.18) has a finite optimal solution taken on at the extreme point x_j^p of X_j, then one checks whether $w_j^p - \pi L_j^p - \alpha_j > 0$. If so, the column $(L_j^p, 0 \ldots 1 \ldots 0)$ is added to the restricted master problem. This column obviously also has $m+n$ components, the first m of which are given by L_j^p. The last n components are all zero except for the jth, which is equal to unity. If $w_j^p - \pi L_j^p - \alpha_j \leq 0$, then no column is added to the restricted master problem for that particular index j. If, moreover, all the problems (3.18) have finite optimal solutions x_j^p $(j = 1 \ldots n)$ for which $w_j^p - \pi L_j^p - \alpha_j \leq 0$, then the algorithm stops—the current optimal solution to the restricted master problem is also optimal for the full master problem.

It can now obviously happen that several columns (but at most n) are added to the restricted master problem in any one iteration. It is also obvious that the method outlined in section 3.3.2 is a special case of the method here—the case for $n = 1$.

A bound for the optimal solution value for the extremal problem (3.17) [or, equivalently, the original problem (3.16)] can be obtained here, too. Suppose that, at some iteration t, the current optimal solution value for the restricted master problem is z_t. Let π and $\alpha = (\alpha_1 \ldots \alpha_n)$, as before, be optimal dual multipliers associated with the restricted master problem. If, for one or more of the indices $j = 1 \ldots n$, the corresponding problems (3.18) have unbounded optimal solutions in the current iteration, then no bound on $\bar{z}$, the true optimal solution value, can be obtained. However, if for all the indices j, the subproblems (3.18) have finite optimal solutions x_j^p, an upper bound is given by:

$$\bar{z} \leq z_t + \sum_{j=1}^{n} (w_j^p - \pi L_j^p - \alpha_j).$$

Here we obtain an *upper* bound, since problem (3.16) involves maximization.

For a numerical example of the application of the Dantzig–Wolfe decomposition method to a block-angular problem, consider the following:

$$\begin{aligned}
&\text{Maximize} && 3x_{11} + x_{21} + 2x_{12} + x_{22} \\
&\text{s.t.:} && 2x_{11} + \quad + x_{12} + 2x_{22} \leq 6, \\
& && x_{11} + \; x_{21} \quad + \; x_{22} \leq 3, \\
& && x_{11} + 2x_{21} \qquad\quad \leq 1, \\
& && \qquad\quad x_{12} - \; x_{22} \leq 1, \\
& && x_{11}, x_{21}, x_{12}, x_{22} \geq 0.
\end{aligned} \tag{3.19}$$

This is obviously a block-angular problem of the same type as (3.16). Here, $n = 2$, and the correspondence between (3.16) and (3.19) is brought out by the following identifications:

$$c_1 = (3, 1); \qquad c_2 = (2, 1);$$

$$A_1 = \begin{bmatrix} 2 & 0 \\ 1 & 1 \end{bmatrix}; \qquad A_2 = \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix}; \qquad a = \begin{bmatrix} 6 \\ 3 \end{bmatrix};$$

$$B_1 = (1, 2); \qquad b_1 = 1;$$

$$X_1 = \{(x_{11}, x_{21}) \mid x_{11} + 2x_{21} \leq 1,\ x_{11}, x_{21} \geq 0\};$$

$$B_2 = (1, -1); \qquad b_2 = 1;$$

$$X_2 = \{(x_{12}, x_{22}) \mid x_{12} - x_{22} \leq 1,\ x_{12}, x_{22} \geq 0\}.$$

To start off the algorithm, one needs some columns pertaining to both sets X_1 and X_2 to form a feasible solution for the first restricted master problem. In this

case, (0, 0) is an obvious extreme point of both X_1 and X_2. The associated columns for the restricted master problem are (0, 0, 1, 0) and (0, 0, 0, 1). The associated objective function coefficients are both zero. One may hence write the restricted master problem:

$$\begin{aligned} &\text{Maximize} && 0\lambda_1^1 + 0\lambda_2^1 \\ &\text{s.t.:} && \begin{bmatrix}0\\0\end{bmatrix}\lambda_1^1 + \begin{bmatrix}0\\0\end{bmatrix}\lambda_2^1 \leq \begin{bmatrix}6\\3\end{bmatrix}, \\ & && 0 \leqq \lambda_2^1 = 1, \\ & && 0 \leq \lambda_2^1 = 1. \end{aligned}$$

Iteration 1. The restricted master problem now has exactly one feasible, and hence optimal, solution: $\lambda_1^1 = \lambda_2^1 = 1$. All dual multipliers are zero. Two problems of type (3.18) must now be solved:

For $j = 1$:		For $j = 2$:	
Maximize	$3x_{11} + x_{21}$	Maximize	$2x_{12} + x_{22}$
s.t.:	$x_{11} + 2x_{21} \leq 1,$	s.t.:	$x_{12} - x_{22} \leq 1,$
	$x_{11}, x_{21} \geq 0.$		$x_{12}, x_{22} \geq 0.$

The first problem, for $j = 1$, has a finite solution, the extreme point $x_1^2 = (1, 0)$ of the set X_1. The associated column for the restricted master problem is (2, 1, 1, 0), and the objective function coefficient is 3. Since $\alpha_1 = 0$ and $\pi = 0$ in the current iteration, this column is added to the restricted master problem. The second problem has an unbounded solution, tending to $+\infty$ in solution value along the extreme ray $k\tilde{x}_2^1$, $k \geq 0$ and $\tilde{x}_2^1 = (1, 1)$ of the set X_2. The associated column is (3, 1, 0, 0), and the objective function coefficient for the restricted master problem is 3. Thus, two new columns have been identified, and the restricted master problem gets extended accordingly. Since the subproblem above for $j = 2$ has an unbounded optimal solution, no upper bound on $\bar{z}$, the optimal solution value of the original problem (3.19), is obtained in this iteration.

Iteration 2. The restricted master problem is now:

$$\begin{aligned} &\text{Maximize} && 0\lambda_1^1 + 3\lambda_1^2 + 0\lambda_2^1 + 3\delta_2^1 \\ &\text{s.t.:} && \begin{bmatrix}0\\0\end{bmatrix}\lambda_1^1 + \begin{bmatrix}2\\1\end{bmatrix}\lambda_1^2 + \begin{bmatrix}0\\0\end{bmatrix}\lambda_2^1 + \begin{bmatrix}3\\1\end{bmatrix}\delta_2^1 \leq \begin{bmatrix}6\\3\end{bmatrix}, \\ & && \lambda_1^1 + \lambda_1^2 = 1, \\ & && \lambda_2^1 = 1, \\ & && \lambda_1^1, \lambda_1^2, \lambda_2^1, \delta_2^1 \geq 0. \end{aligned}$$

The optimal solution is $\lambda_1^1 = 0$, $\lambda_1^2 = 1$, $\lambda_2^1 = 1$, $\delta_2^1 = 4/3$. The optimal dual multipliers are $\pi = (\pi_1, \pi_2) = (1, 0)$, $\alpha_1 = 1$, $\alpha_2 = 0$. The optimal solution value z_2 is 7. Again, two subproblems must now be solved to generate additional columns, or to conclude that the current optimal solution to the restricted master problem is also optimal for the full master problem:

For $j = 1$:		For $j = 2$:	
Maximize	$x_{11} + x_{21}$	Maximize	$x_{12} - x_{22}$
s.t.:	$x_{11} + 2x_{21} \leq 1$,	s.t.:	$x_{12} - x_{22} \leq 1$,
	$x_{11}, x_{21} \geq 0$,		$x_{12}, x_{22} \geq 0$.

The first subproblem has the optimal extreme point $x_1^2 = (1, 0)$ of the set X_1. Since this extreme point has been used earlier to create a column for the restricted master problem, it must obviously hold that $c_1 x_1^2 - \pi A_1 x_1^2 - \alpha_1 \leq 0$. Therefore, no new column is added to the restricted master problem for $j = 1$. For $j = 2$, there are many optimal solutions, but one is given by the extreme point $x_2^2 = (1, 0)$ of the set X_2. Since $c_2 x_2^2 - \pi A_2 x_2^2 - \alpha_2 = 2 - 1 - 0 > 0$, the associated column $(1, 0, 0, 1)$ is added to the restricted master problem. The objective function coefficient is 2. An upper bound on $\bar{z}$ may now be computed. It is given by $z_2 + 0 + 1 = 8$.

Iteration 3. The restricted master problem is:

$$\text{Maximize} \quad 0\lambda_1^1 + 3\lambda_1^2 + 0\lambda_2^1 + 2\lambda_2^2 + 3\delta_2^1$$

$$\text{s.t.:} \quad \begin{bmatrix} 0 \\ 0 \end{bmatrix} \lambda_1^1 + \begin{bmatrix} 2 \\ 1 \end{bmatrix} \lambda_1^2 + \begin{bmatrix} 0 \\ 0 \end{bmatrix} \lambda_2^1 + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \lambda_2^2 + \begin{bmatrix} 3 \\ 1 \end{bmatrix} \delta_2^1 \leq \begin{bmatrix} 6 \\ 3 \end{bmatrix},$$

$$\lambda_1^1 + \lambda_1^2 = 1,$$

$$\lambda_2^1 + \lambda_2^2 = 1,$$

$$\lambda_1^1, \lambda_1^2, \lambda_2^1, \lambda_2^2, \delta_2^1 \geq 0.$$

The optimal solution is $\lambda_1^1 = \lambda_2^1 = 0$, $\lambda_1^2 = \lambda_2^2 = 1$, $\delta_2^1 = 1$. The optimal dual multipliers are $\pi = (\pi_1, \pi_2) = (1, 0)$, $\alpha_1 = \alpha_2 = 1$. The optimal solution value is 8. Since the multiplier vector $\pi = (1, 0)$ is the same as in the previous iteration, it is clear that no new columns to add to the restricted master problem will be found by solving the subproblems (3.18) for $j = 1, 2$. Hence the computations stop at this point—the current optimal restricted master problem solution is also optimal for the full master problem. An optimal solution to the original problem (3.19) may be recovered as:

$$(x_{11}, x_{21}) = 1x_1^2 = (1, 0),$$

$$(x_{12}, x_{22}) = 1x_2^2 + 1\bar{x}_2^1 = (2, 1).$$

Therefore the complete optimal solution is $(x_{11}, x_{21}, x_{12}, x_{22}) = (1, 0, 2, 1)$.

The Dantzig–Wolfe method, applied to a block-angular LP problem, is again a clear case of a two-level methodology. The restricted master problem is the supremal subproblem, and *the infimal subproblems in the adjustment phase* are of the same kind as (3.18). In the adjustment phase, simplex multipliers are transmitted to the infimal subproblems from the supremal one, and the infimal subproblems transmit back candidate columns for the supremal subproblem. In the execution phase, a solution to the original problem can be obtained by means of relations like (3.6) (one such relation for each infimal subproblem). That is, suppose $\bar{\lambda}_j^p$ and $\bar{\delta}_j^r$ denote a solution to the extremal problem (optimal or merely satisfactory but nonoptimal). A solution to the original problem (3.16) can then be obtained as $\bar{x} = (\bar{x}_1, \bar{x}_2 \ldots \bar{x}_n)$, where $\bar{x}_j = \sum_p \bar{\lambda}_j^p x_j^p + \sum_r \bar{\delta}_j^r \tilde{x}_j^r$. However, in actual computer codes, this manner of recovering a solution to the original problem in the execution phase is usually not to be recommended. Instead, a solution to the original problem is recovered by means of *infimal subproblems in the execution phase* of the kind (for $j = 1 \ldots n$):

$$
\begin{aligned}
&\text{Maximize} && c_j x_j \\
&\text{s.t.:} && A_j x_j \leq \textstyle\sum_p L_j^p \bar{\lambda}_j^p + \sum_r \tilde{L}_j^r \bar{\delta}_j^r, \\
& && B_j x_j \leq b_j, \\
& && x_j \geq 0.
\end{aligned}
$$

Let $\bar{x}_j$ be an optimal solution. Then $\bar{x} = (\bar{x}_1, \bar{x}_2 \ldots \bar{x}_n)$. This means that the overall solution to the original problem is obtained in the execution phase from the infimal subproblems. However, the infimal subproblems in the execution phase are different from the infimal subproblems in the adjustment phase. This possibility was mentioned earlier (section 2.1.3); for further discussion, see section 4.2.

The original problem (3.16) is often taken to represent a planning problem in a divisionally organized firm with n divisions (or departments). The vector x_j gives activity levels for the jth division. The divisions are independent, except that they jointly utilize certain scarce resources, m in number, the availability of which is given by the vector a. The coupling constraints $\sum_{j=1}^n A_j x_j \leq a$ express this joint resource usage. Additionally, each division faces local constraints, pertaining only to its own activities, given by $B_j x_j \leq b_j$, $x_j \geq 0$. In this situation, the Dantzig–Wolfe method has a well-known economic interpretation (see, for instance, Almon 1963; Baumol and Fabian 1964; Bagrinowski 1975, pp. 147–150; Mandel' 1973). The supremal subproblem pertains to corporate headquarters. There are n infimal subproblems, one for each division, of the form (3.18). The supremal subproblem iteratively sends a price vector π associated with the joint resources to the infimal subproblems, and the infimal subproblems respond with plan proposals. These plan

proposals are of the form $(A_j x_j^p)$ or $(A_j \hat{x}_j^r)$ and may be considered as vectors of the joint resources demanded, given the announced prices π. It has been suggested that the Dantzig–Wolfe method could actually be implemented as an institutionalized planning procedure in corporations. This matter is explored further in Chapter 6.

3.4 THE DANTZIG–WOLFE METHOD FOR NONLINEAR PROGRAMS

Decomposition methods for nonlinear mathematical programming problems have also been developed. In this section we present a method that is based on the Dantzig–Wolfe decomposition principle for linear programs. It aims at solving nonlinear problems that are generalizations of the block-angular structures discussed in section 3.3.6. For an original statement of the method, we refer to Dantzig (1963, Chapter 24); Lasdon (1970, pp. 242–254) and Sekine (1963) are also of interest.

We will consider the following optimization problem, the original problem of this section:

$$\begin{aligned} &\text{Maximize} && \sum_{j=1}^{n} f_j(x_j) \\ &\text{s.t.:} && \sum_{j=1}^{n} A_j x_j \leq a, \\ & && x_j \in X_j \qquad \text{for } j = 1 \ldots n, \end{aligned} \tag{3.20}$$

where each x_j is an n_j vector, a is a column vector of dimension m, and each A_j is an $m \times n_j$ matrix. Each function f_j is continuous and concave, and each set X_j is closed, bounded, and convex. We will also assume that the set of feasible solutions to (3.20) is nonempty. Since the sets X_j are not necessarily polyhedral, we cannot make use of the characterization result of section 3.3.1. Instead, we shall make use of *grid linearization*, a technique introduced by Wolfe (1967).

For any one of the convex sets X_j $(j = 1 \ldots n)$, consider a set of *grid points* $\{\hat{x}_j^s \mid \hat{x}_j^s \in X_j,\ s = 1 \ldots S(j)\}$. Their convex hull is

$$\hat{X}_j = \left\{ x_j \middle| x_j = \sum_{j=1}^{S(j)} \lambda_j^s \hat{x}_j^s, \qquad \lambda_j^s \geq 0 \qquad \text{for } s = 1 \ldots S(j), \sum \lambda_j^s = 1 \right\}.$$

Clearly, $\hat{X}_j \subseteq X_j$ by convexity. We can now formulate the following linear program (3.21), which approximates (3.20):

$$\begin{aligned} \text{Maximize} \quad & \sum_{j=1}^{n} \sum_{s=1}^{S(j)} f_j(\hat{x}_j^s)\lambda_j^s \\ \text{s.t.:} \quad & \sum_{j=1}^{n} \sum_{s=1}^{S(j)} (A_j\hat{x}_j^s)\lambda_j^s \leq a, \\ & \sum_{s=1}^{S(j)} \lambda_j^s = 1 \qquad \text{for } j = 1 \ldots n, \\ & \lambda_j^s \geq 0. \end{aligned} \tag{3.21}$$

(3.21) is an approximation of (3.20) in the following sense: A feasible solution $\{\bar{\lambda}_j^s\}$ to (3.21) determines $\bar{x}_j = \sum \bar{\lambda}_j^s \hat{x}_j^s$, $j = 1 \ldots n$, as a feasible solution to (3.20). The objective function value of (3.21) gives a lower bound to (3.20), since (by concavity)

$$\sum_{j=1}^{n} \sum_{s=1}^{S(j)} f_j(\hat{x}_j^s)\bar{\lambda}_j^s \leq \sum_{j=1}^{n} f_j\Big(\sum_{s=1}^{S(j)} \bar{\lambda}_j^s \hat{x}_j^s\Big).$$

Problem (3.21) is the restricted master problem, or supremal subproblem, in the Dantzig–Wolfe method for nonlinear programs.

Given the convexity assumptions, one can imagine that, for a suitable choice of grid points, (3.21) would constitute a close approximation to (3.20). A suitable choice of grid points is achieved in the following algorithm in an iterative fashion by means of a column generation scheme. This scheme will generate a close approximation in the sense that one is able to show convergence to an optimal solution to (3.20) in an infinite number of iterations. This implies that after sufficiently many iterations one can come arbitrarily close to an optimal solution.

Suppose that, at some iteration, one has available as grid points the sets $\{\hat{x}_j^s | s = 1 \ldots S(j)\}$, resulting in a feasible restricted master problem (3.21), and that one solves (3.21) using those grid points. Let the m vector π, associated with the m inequality constraints of (3.21), and the n vector α, associated with the n equality constraints of (3.21), be optimal dual multipliers. One then solves the following programming problem for each index $j = 1 \ldots n$:

$$\begin{aligned} \text{Maximize} \quad & f_j(x_j) - \pi A_j x_j \\ \text{s.t.:} \quad & x_j \in X_j. \end{aligned} \tag{3.22}$$

Problems (3.22), for $j = 1 \ldots n$, are infimal subproblems. They may be hard to solve, but we will not go into that matter here. Let x_j' denote an optimal solution to (3.22). If $f_j(x_j') - \pi A_j x_j' - \alpha_j > 0$, x_j' is added as a grid point, and hence the column $(A_j x_j', 0 \ldots 1 \ldots 0)$ with objective function coefficient $f_j(x_j')$ is added to the restricted master problem (3.21). If $f_j(x_j') - \pi A_j x_j' - \alpha_j \leq 0$

for all j, the algorithm stops. An optimal solution to (3.20) can then be obtained from the solution to (3.21) in the last iteration.

A proof of the optimality condition may be found in Dantzig (1963, pp. 475–476). The above method converges only in infinitely many iterations, which is not the case with linear programs (Dantzig 1963, pp. 476–478). However, just as in the linear case, a termination criterion based on the existence of upper and lower bounds at each iteration may be included. To obtain an initial solution to the restricted master problem (3.21), artificial columns may be used. In the adjustment phase, then, (3.21) is the supremal subproblem, and the infimal subproblems are of the type (3.22). The infimal subproblems are iteratively supplied with dual multipliers from the supremal subproblem, and transmit back columns. In the execution phase, a solution to the original problem can be obtained from the last solution to the supremal subproblem, denoted $\{\bar{\lambda}_j^s\}$, by the relations $\bar{x}_j = \sum \bar{\lambda}_j^s \hat{x}_j^s$ $(j = 1 \ldots n)$.

The Dantzig–Wolfe decomposition method for linear block-angular problems is a special case of the above algorithm. In fact, assume each f_i to be linear, each X_i polyhedral, and take the respective extreme points as grid points. Then grid linearization corresponds to the Dantzig–Wolfe decomposition method for linear programs. There is, however, a theoretical difference. In the case of linear models, it is optional to retain columns that are priced out of the basis of the restricted master problem at a particular iteration (see section 4.2 for further discussion). In the nonlinear case, the situation is more complex. In general, columns have to be retained to ensure convergence. Murphy (1973) gives two sufficient conditions under which columns may be dropped.

The Dantzig–Wolfe decomposition method for nonlinear programs is utilized in Chapters 9 and 10 (an electricity generation problem and a problem regarding water pollution control). The first case involves only one infimal subproblem, and the second case involves several.

3.5 THE BENDERS ALGORITHM AND SOME EXTENSIONS

3.5.1 AN OUTLINE OF THE BENDERS ALGORITHM

Benders' algorithm (Benders 1962) was originally developed for mixed-integer linear programming problems. The development of the Benders algorithm is the subject of this section. We discuss here optimization problems of the following type:

$$\begin{aligned} &\text{Minimize} && cx + f(y) \\ &\text{s.t.:} && Ax + F(y) \geq b, \\ & && y \in Y, \\ & && x \geq 0. \end{aligned} \tag{3.23}$$

Problem (3.23) is the original, or overall, problem of this section. Here, two types of variables are considered, n "linear" variables given by the vector x and q "special" variables represented by y. In (3.23), c, b, A are, respectively, an n vector, m vector, and an $m \times n$ matrix; $f(y)$ and $F(y)$ are, respectively, a real-valued continuous function and an m-dimensional vector-valued continuous function defined on a compact set Y, a subset of E^q. Problem (3.23) is not necessarily a linear one. The functions $f(y)$ and $F(y)$ could, for instance, be linear, but the set Y could be a set of integer-valued vectors. In that case, (3.23) is a mixed-integer program. We have already noted that Benders' algorithm was originally developed precisely for this type of integer programming problems. On the other hand, with suitable specifications on $f(y)$, $F(y)$, and Y, (3.23) can be a linear programming problem.

We will now reformulate problem (3.23). Our objective is to derive an equivalent problem formulation, in somewhat the same fashion as in the discussion of the Dantzig–Wolfe decomposition method. There, we started with an LP problem (3.4) and then derived the equivalent extremal problem (3.7).

In (3.23), the feasible choices of y are limited by the constraint $y \in Y$. However, if y is fixed, an ordinary linear program results:

$$\begin{aligned} &\text{Minimize} \quad cx \\ &\text{s.t.:} \quad Ax \geq b - F(y), \\ &\qquad\quad x \geq 0. \end{aligned} \tag{3.24}$$

For certain choices of y, (3.24) may not possess feasible solutions. Hence, y must be chosen subject to the following constraint (in addition to $y \in Y$):

$$y \in \mathcal{Y} = \{y \mid \text{there exists } x \geq 0 \text{ such that } Ax \geq b - F(y)\}.$$

From Farkas's lemma* (see, e.g., Gale 1960, pp. 42–49), it follows that there either exists an x such that

$$Ax \geq b - F(y), \qquad x \geq 0$$

or there exists a u such that

$$uA \leq 0, \qquad u(b - F(y)) > 0, \qquad u \geq 0,$$

but not both. Hence, if y is chosen such that $u(b - F(y)) \leq 0$ for all u satisfying $uA \leq 0$, $u \geq 0$, then $y \in \mathcal{Y}$. Let $\tilde{u}^1 \ldots \tilde{u}^R$ be the finite set of extreme rays of the cone $\{u \mid uA \leq 0, u \geq 0\}$. It then holds that $u(b - F(y)) \leq 0$ for all u satisfying $uA \leq 0$, $u \geq 0$, if and only if $\tilde{u}^r(b - F(y)) \leq 0$ for all extreme rays $\tilde{u}^r$ ($r = 1 \ldots R$). The set $\mathcal{Y}$ may hence be specified as

$$\mathcal{Y} = \{y \mid \tilde{u}^r(b - F(y)) \leq 0,\ r = 1 \ldots R\}.$$

* *Farkas's lemma*: Either there does exist a nonnegative solution to the system of linear inequalities $Ax \geq d$, or the system $uA \leq 0$, $ud > 0$, has a nonnegative solution.

We may rewrite (3.23) as

$$\begin{aligned} &\text{Minimize} && \{f(y)+\min[cx \mid Ax \geq b-F(y),\ x \geq 0]\} \\ &\text{s.t.:} && y \in Y \cap \mathcal{Y}. \end{aligned} \tag{3.25}$$

Analyzing (3.25), we can now distinguish two cases.

Case 1: $Y \cap \mathcal{Y} \neq \emptyset$. Feasible solutions to (3.25) exist, which also means that (3.23) has a feasible solution. The optimal solution value of (3.23) could, however, be unbounded. For $y \in Y \cap \mathcal{Y}$, the inner minimization is a linear programming problem. This problem either has a finite optimal solution, or an unbounded solution, in which case the optimal solution value is taken as $-\infty$. Its dual is

$$\begin{aligned} &\text{Maximize} && u(b-F(y)) \\ &\text{s.t.:} && uA \leq c,\ u \geq 0. \end{aligned}$$

The dual then has either a finite optimal solution or no solution at all. In the latter case, the optimal solution value is taken as $-\infty$. It then holds that

$$\min\{cx \mid Ax \geq b-F(y),\ x \geq 0\} = \max\{u(b-F(y)) \mid uA \leq c,\ u \geq 0\},$$

regardless of whether the minimization problem has a finite optimum solution or an unbounded solution. We may then rewrite (3.25) as

$$\begin{aligned} &\text{Minimize} && \{f(y)+\max[u(b-F(y)) \mid uA \leq c,\ u \geq 0]\} \\ &\text{s.t.:} && y \in Y \cap \mathcal{Y}. \end{aligned} \tag{3.26}$$

If the inner maximization is not feasible, then (3.26), and consequently (3.23) as well, have solution values tending to $-\infty$. If the inner maximization is feasible, the maximum value is taken on at an extreme point u^p of the constraint set $\{u \mid uA \leq c,\ u \geq 0\}$. Let there be P such extreme points. We may then rewrite (3.26) as

$$\begin{aligned} &\text{Minimize} && \{f(y)+\max_{1 \leq p \leq P}[u^p(b-F(y))]\} \\ &\text{s.t.:} && y \in Y \cap \mathcal{Y}. \end{aligned} \tag{3.27}$$

If there exists no u such that $uA \leq c$, $u \geq 0$, then $P = 0$, i.e., the set of extreme points is empty. In that case, the inner maximum is taken as $-\infty$, so formulation (3.27) is valid. Formulation (3.27) may be rewritten further as:

$$\begin{aligned} &\text{Minimize} && z \\ &\text{s.t.:} && z \geq f(y)+u^p(b-F(y)) \qquad (p=1 \ldots P), \\ &&& \tilde{u}^r(b-F(y)) \leq 0 \qquad (r=1 \ldots R), \\ &&& y \in Y. \end{aligned} \tag{3.28}$$

Case 2: $Y \cap \mathcal{Y} = \varnothing$. In this case, (3.23) has no feasible solution, which implies that (3.28) also has no feasible solution. Conversely, if (3.28) has no feasible solution, then there are no feasible choices of y in (3.23), so (3.23) is also infeasible.

From the previous development, it is clear that the following statements must hold:

1. Problem (3.23) has a feasible solution if and only if (3.28) has a feasible solution.
2. Problem (3.23) has an unbounded solution value if and only if (3.28) has an unbounded solution value.
3. If (z^o, y^o) is an optimal solution to (3.28) and x^o an optimal solution to the problem

$$\begin{aligned} &\text{Minimize} && cx \\ &\text{s.t.:} && Ax \geq b - F(y^o), \\ & && x \geq 0, \end{aligned}$$

then (x^o, y^o) is an optimal solution to (3.23). Conversely, if (x^o, y^o) is an optimal solution to (3.23), set $z^o = cx^o + f(y^o)$, and (z^o, y^o) is optimal for (3.28).

Problem (3.28) is the desired equivalent formulation of (3.23). It plays the same role in the Benders algorithm as the full master problem in the decomposition principle of Dantzig and Wolfe. If the complete specification of (3.28) were available, it could be solved right away. However, it may have a very large number of restrictions, because the number of extreme points (P) and extreme rays (R) could be very large. One method of solving (3.28) is to generate the constraints successively, as they are needed, which is precisely what Benders algorithm accomplishes. This is, again, similar to the situation in the Dantzig–Wolfe algorithm, where the extremal problem has many columns, but where these are generated successively as the algorithm proceeds.

More precisely, suppose one has at hand the following restricted version of problem (3.28), where only a subset $\mathcal{P}$ of the restrictions $z \geq f(y) + u^p(b - F(y))$ and a subset $\mathcal{R}$ of the restrictions $\tilde{u}^r(b - F(y)) \leq 0$ are given:

$$\begin{aligned} &\text{Minimize} && z \\ &\text{s.t.:} && z \geq f(y) + u^p(b - F(y)) \qquad (p \in \mathcal{P}), \\ & && \tilde{u}^r(b - F(y)) \leq 0 \qquad (r \in \mathcal{R}), \\ & && y \in Y. \end{aligned} \tag{3.29}$$

Let $(\bar{z}, \bar{y})$ be an optimal solution. Then consider the LP problem:

$$\begin{aligned} &\text{Maximize} && u(b - F(\bar{y})) \\ &\text{s.t.:} && uA \leq c, \\ & && u \geq 0, \end{aligned} \tag{3.30}$$

which is, of course, the dual of $\min \{cx \mid Ax \geq b - F(\bar{y}),\ x \geq 0\}$. If (3.30) has an unbounded optimal solution value, then some constraint $\tilde{u}^{r'}(b - F(y)) \leq 0$ must be violated for the extreme ray $\tilde{u}^{r'}$ and for $y = \bar{y}$. In that case, that extreme ray is identified and the corresponding constraint is added to the problem (3.29). If (3.30) has a finite optimum, given by the extreme point $u^{p'}$, then it may be that the corresponding constraint $z \geq f(y) + u^{p}(b - F(y))$ does not hold for that index p and for $y = \bar{y}$, $z = \bar{z}$. In that case, too, the corresponding constraint is added to (3.29). However, suppose that $\bar{z} \geq f(\bar{y}) + u^{p'}(b - F(\bar{y}))$ does hold, where $u^{p'}$ is an optimal extreme point solution to (3.30). It must then hold that $\bar{z} - f(\bar{y}) = u^{p'}(b - F(\bar{y}))$. If not, assume $\bar{z} - f(\bar{y}) > u^{p'}(b - F(\bar{y}))$. In view of (3.30), $u^{p'}(b - F(\bar{y})) \geq \max_{p \in \mathcal{P}} \{u^{p}(b - F(\bar{y}))\}$. Hence, if $\bar{z} - f(\bar{y}) > u^{p'}(b - F(\bar{y}))$, then $\bar{z} - f(\bar{y}) > \max_{p \in \mathcal{P}} \{u^{p}(b - F(\bar{y}))\}$. This contradicts the optimality of $(\bar{z}, \bar{y})$ for (3.29), so it must be that $\bar{z} - f(\bar{y}) = u^{p'}(b - F(\bar{y}))$.

Summing up, if (3.30) has a finite optimal solution and if $\max \{u(b - F(\bar{y})) \mid uA \leq c,\ u \geq 0\} = \bar{z} - f(\bar{y})$, then the solution $(\bar{z}, \bar{y})$ is feasible and optimal in (3.28). Optimality follows since (3.28) and (3.29) have the same objective function; however, the feasible region of (3.28) is included in that of (3.29).

The steps of the Benders algorithm can now be outlined as follows:

Step 0. Initiate the algorithm with a problem of the type (3.29), where the sets $\mathcal{P}$ and $\mathcal{R}$ may contain very few elements (or even none at all).

Step 1. Solve the current version of problem (3.29), with those constraints that are presently at hand. If (3.29) has no feasible solution, then stop; the original problem (3.23) also has no feasible solution. If (3.29) has a bounded optimal solution $(\bar{z}, \bar{y})$, go on to Step 2. If (3.29) has an unbounded solution, let $(\bar{z}, \bar{y})$ be any feasible solution such that $\bar{z} = -\infty$, and go on to Step 2.

Step 2. Solve problem (3.30), with $\bar{y}$ specified in Step 1. If (3.30) is infeasible, stop; the original problem (3.23) is either infeasible or has unbounded optimal solutions. (This situation can arise only in the first iteration and is discussed in the following subsection.) If (3.30) has an infinite optimal solution value, go to Step 3. If (3.30) has a finite optimal solution, go on to Step 4.

Step 3. In this case, $u(b - F(\bar{y}))$ tends to $+\infty$ along some extreme ray $\tilde{u}^{r}$. Add the corresponding constraint $\tilde{u}^{r}(b - F(y)) \leq 0$ to problem (3.29) and return to Step 1.

Step 4. In this case, let u^{p} be an optimal extreme point solution. If $u^{p}(b - F(\bar{y})) = \bar{z} - f(\bar{y})$, then stop—the optimal solution $(\bar{z}, \bar{y})$ to (3.29) is also optimal

for (3.28). If $u^p(b-F(\bar{y}))>\bar{z}-f(\bar{y})$, the corresponding constraint $z \geq f(y)+u^p(b-F(y))$ is added to problem (3.29). Go back to Step 1.

An iteration begins with Step 1 and ends upon return to that step. Finite convergence is guaranteed, since there are only a finite number of extreme points (P) and extreme rays (R), and one extreme point or extreme ray is identified at each iteration.

The Benders algorithm, like the Dantzig–Wolfe decomposition algorithm, produces a lower bound on the value of the optimal solution to the original problem (3.23) as it proceeds. Let (z_t, y_t) be an optimal solution to (3.29) at some iteration t. Then, obviously, z_t is a lower bound on the optimal solution value for (3.23), since the feasible region of (3.28) is included in that of (3.29). This lower bound z_t converges monotonically on the true optimal solution value. If problem (3.30) has a finite optimal solution for $y = y_t$, then obviously $f(y_t)+\max\{u(b-F(y_t))\,|\,uA \leq c, u \geq 0\}$ is an *upper* bound on the true optimal value. This upper bound, though, does *not* converge monotonically. Also, it should be noted that if, in some iteration t, the dual of (3.30) is feasible, then this does not guarantee that it will be feasible in all following iterations.

The Benders algorithm, as outlined here, can clearly be regarded as a two-level method. Problem (3.29) is the supremal subproblem. There is only one infimal subproblem, (3.30). The information exchange between supremal and infimal subproblem in the adjustment phase was described in the preceding outline of the steps of the algorithm (Steps 1–4). In the execution phase, a solution to the original problem (3.23) is recovered from *both* the supremal and infimal subproblems. The supremal subproblem supplies the y component, the infimal subproblem the x component of the resulting solution to the original problem. The x component is, of course, only an optimal dual solution to the infimal subproblem (3.30). The information exchange between the subproblems in the adjustment phase may be visualized as in Figure 3.6.

3.5.2 A NOTE ON STEP 2 OF THE BENDERS ALGORITHM

In Step 2 of the Benders algorithm, it may happen that problem (3.30) has no feasible solution. The question is then: What conclusion can be drawn about the original problem (3.23)? The answer is that it could be either infeasible or have an unbounded optimal solution. Consider the following example:

$$
\begin{aligned}
\text{Minimize} \quad & -x_1-x_2 \\
\text{s.t.:} \quad & -x_1+x_2-y_1 \geq 0, \\
& x_1-x_2-y_2 \geq 0, \\
& x_1, x_2 \geq 0, \\
& (y_1, y_2) \in Y = \{(y_1, y_2)\,|\,1 \leq y_1 \leq 2,\ 1 \leq y_2 \leq 2\}.
\end{aligned}
\tag{3.31}
$$

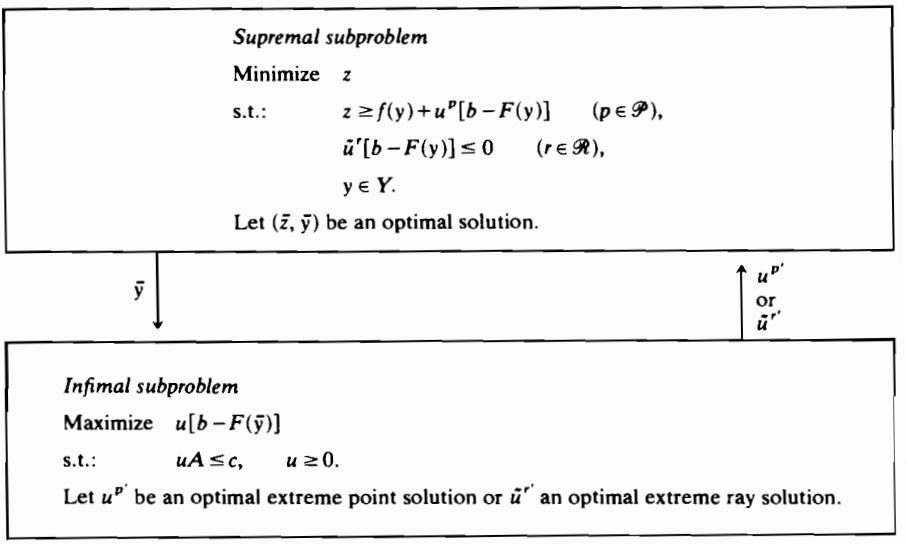


FIGURE 3.6 The adjustment phase of the Benders algorithm.

Suppose one wants to solve this problem (which is not feasible) with the Benders algorithm. To begin with, let the sets $\mathcal{P}$ and $\mathcal{R}$ in (3.29) be empty. In Step 1 of the first iteration, one would hence formulate and solve the following problem, with $\mathcal{P} = \mathcal{R} = \emptyset$:

$$\begin{aligned} &\text{Minimize} && z \\ &\text{s.t.:} && 1 \leq y_1 \leq 2, \\ & && 1 \leq y_2 \leq 2. \end{aligned}$$

Since this problem has optimal solution values tending to $-\infty$, one would in Step 1 pick any feasible (y_1, y_2), for instance, $y_1 = 2$, $y_2 = 2$. In Step 2, one would then solve the following maximization problem, corresponding to (3.30):

$$\begin{aligned} &\text{Maximize} && 2u_1 + 2u_2 \\ &\text{s.t.:} && -u_1 + u_2 \leq -1, \\ & && u_1 - u_2 \leq -1, \\ & && u_1, u_2 \geq 0. \end{aligned} \tag{3.32}$$

Problem (3.32) is not feasible.

Now suppose one changes the set Y in (3.31) to $0 \leq y_1 \leq 2$, $0 \leq y_2 \leq 2$. Then (3.31) has solutions with objective function value tending to $-\infty$. However, in

Step 1 of the first iteration, one may still set $y_1 = 2$, $y_2 = 2$, and one will then in Step 2 obtain problem (3.32), an infeasible problem.

Hence, if problem (3.30) in Step 2 has no feasible solution, a further investigation of the reasons may be necessary. Obviously, this situation must occur in the first iteration, if it is to occur at all. This follows since the feasible set of the infimal subproblem is the same in all iterations (i.e., it does not depend on y).

3.5.3 A NUMERICAL EXAMPLE

Consider the following mixed-integer programming problem taken from Garfinkel and Nemhauser (1972, p. 141), and written in the format of (3.23):

$$\begin{aligned} &\text{Minimize} && 2x_1+6x_2+2y_1+3y_2 \\ &\text{s.t.:} && -x_1+2x_2+3y_1- y_2 \geq 5, \\ & && x_1-3x_2+2y_1+2y_2 \geq 4, \\ & && x_1, x_2 \geq 0, \\ & && y_1, y_2 = 0, 1, \text{ or } 2. \end{aligned} \tag{3.33}$$

Problem (3.33) is the overall problem under consideration. If y_1 and y_2 are fixed, an ordinary LP problem results. The dual of this LP problem has the constraint set

$$\{(u_1, u_2) \mid -u_1+u_2 \leq 2,\ 2u_1-3u_2 \leq 6,\ u_1 \geq 0,\ u_2 \geq 0\}.$$

This constraint set has three extreme points:

$$u^1 = (0, 0); \qquad u^2 = (0, 2); \qquad u^3 = (3, 0);$$

and two extreme rays, which may be written as:

$$\tilde{u}^1 = (1, 1); \qquad \tilde{u}^2 = (3, 2).$$

Hence, (3.33) may be rewritten in the following equivalent form, as the full master problem of the Benders algorithm, problem (3.28) above:

$$\begin{aligned} &\text{Minimize} && z \\ &\text{s.t.:} && z \geq 2y_1+3y_2+(0, 0)\begin{bmatrix} 5-3y_1+ y_2 \\ 4-2y_1-2y_2 \end{bmatrix}, \\ & && z \geq 2y_1+3y_2+(0, 2)\begin{bmatrix} 5-3y_1+ y_2 \\ 4-2y_1-2y_2 \end{bmatrix}, \\ & && z \geq 2y_1+3y_2+(3, 0)\begin{bmatrix} 5-3y_1+ y_2 \\ 4-2y_1-2y_2 \end{bmatrix}, \end{aligned}$$

$$(1, 1)\begin{bmatrix}5-3y_1+ y_2\\ 4-2y_1-2y_2\end{bmatrix}\leq 0,$$

$$(3, 2)\begin{bmatrix}5-3y_1+ y_2\\ 4-2y_1-2y_2\end{bmatrix}\leq 0,$$

$y_1, y_2 = 0, 1$, or 2.

Simplifying, this problem may be rewritten as

$$\begin{aligned}
&\text{Minimize} && z\\
&\text{s.t.:} && z \geq 2y_1+3y_2,\\
&&& z \geq 8-2y_1-y_2,\\
&&& z \geq 15-7y_1+6y_2,\\
&&& 9-5y_1-y_2\leq 0,\\
&&& 23-13y_1-y_2\leq 0,\\
&&& y_1, y_2 = 0, 1, \text{ or } 2.
\end{aligned} \tag{3.34}$$

Problem (3.34) is equivalent to (3.33). Problem (3.34) has the optimal solution $\bar{y}_1 = 2$, $\bar{y}_2 = 0$, $\bar{z} = 4$. To obtain the optimal solution to (3.33), one fixes $y_1 = 2$ and $y_2 = 0$ and solves the resulting LP problem (or its dual), to obtain the optimal solution $\bar{x}_1 = \bar{x}_2 = 0$. (0, 0, 2, 0) is hence the optimal solution to the overall problem (3.33).

Suppose, however, that one desires to solve (3.33) by means of Benders' algorithm. To start out, suppose no extreme points or extreme rays have been generated so far. The initial supremal subproblem is hence

$$\begin{aligned}
&\text{Minimize} && z\\
&\text{s.t.:} && y_1, y_2 = 0, 1, \text{ or } 2.
\end{aligned}$$

Iteration 1. The initial supremal subproblem has unbounded solutions. Hence we set $(\bar{z}, \bar{y}_1, \bar{y}_2)$ arbitrarily to $(-\infty, 1, 1)$. We then construct the infimal subproblem, corresponding to (3.30) above:

$$\begin{aligned}
&\text{Maximize} && 3u_1\\
&\text{s.t.:} && -u_1+ u_2\leq 2,\\
&&& 2u_1-3u_2\leq 6,\\
&&& u_1, u_2\geq 0.
\end{aligned}$$

This problem has an unbounded optimal solution value, going to $+\infty$ along the extreme ray (1, 1) (for example). Hence a corresponding restriction is added to the supremal subproblem.

Iteration 2. The supremal subproblem is now

$$\begin{aligned} &\text{Minimize} && z \\ &\text{s.t.:} && 9-5y_1-y_2 \leq 0, \\ & && y_1, y_2 = 0, 1, \text{ or } 2. \end{aligned}$$

Again, the supremal subproblem has unbounded solutions. Arbitrarily, let $(\bar{z}, \bar{y}_1, \bar{y}_2) = (-\infty, 2, 0)$. The infimal subproblem is then

$$\begin{aligned} &\text{Maximize} && -u_1 \\ &\text{s.t.:} && -u_1 + u_2 \leq 2, \\ & && 2u_1 - 3u_2 \leq 6, \\ & && u_1, u_2 \geq 0. \end{aligned}$$

The infimal subproblem now has the optimal extreme point solution $(0, 0)$. The optimality test is not passed, since

$$(0, 0)\begin{bmatrix} -1 \\ 0 \end{bmatrix} > -\infty - 4 = \bar{z} - 2\bar{y}_1.$$

Hence a constraint corresponding to the extreme point $(0, 0)$ is added to the supremal subproblem.

Iteration 3. The supremal subproblem is now

$$\begin{aligned} &\text{Minimize} && z \\ &\text{s.t.:} && z \geq 2y_1 + 3y_2, \\ & && 9-5y_1-y_2 \leq 0, \\ & && y_1, y_2 = 0, 1, \text{ or } 2. \end{aligned}$$

The supremal subproblem now has the optimal solution $(\bar{z}, \bar{y}_1, \bar{y}_2) = (4, 2, 0)$. The infimal subproblem is exactly the same as in the previous iteration, and this time the optimality test is passed. Hence, the adjustment phase stops, and an optimal solution to the original problem may be recovered.

3.5.4 THE APPLICATION OF THE BENDERS ALGORITHM TO BLOCK-ANGULAR STRUCTURES

In section 3.3.6 the block-angular problem (3.16) was formulated. It is rewritten here for convenience:

$$
\begin{aligned}
&\text{Maximize} && c_1x_1 + c_2x_2 + \cdots + c_nx_n \\
&\text{s.t.:} && A_1x_1 + A_2x_2 + \cdots + A_nx_n \leq a, \\
& && B_1x_1 \leq b_1, \\
& && B_2x_2 \leq b_2, \\
& && \cdots \\
& && B_nx_n \leq b_n, \\
& && x_1, x_2 \ldots x_n \geq 0.
\end{aligned}
\tag{3.16}
$$

This problem, as it stands, is not immediately recognizable as suitable for an application of the Benders algorithm. That is, there is no obvious partitioning of the total set of variables, corresponding to the x and y in (3.23). However, suppose one rewrites the block-angular problem as follows:

$$
\begin{aligned}
&\text{Maximize} && c_1x_1 + c_2x_2 + \cdots + c_nx_n \\
&\text{s.t.:} && A_1x_1 - a_1 \leq 0, \\
& && B_1x_1 \leq b_1, \\
& && A_2x_2 - a_2 \leq 0, \\
& && B_2x_2 \leq b_2, \\
& && \cdots \\
& && A_nx_n - a_n \leq 0, \\
& && B_nx_n \leq b_n, \\
& && a_1 + a_2 + \cdots + a_n = a, \\
& && x_1, x_2 \ldots x_n \geq 0.
\end{aligned}
\tag{3.35}
$$

It is easy to see that

1. Problem (3.16) has no solution if and only if (3.35) has no solution.
2. Problem (3.16) has unbounded optimal solutions if and only if (3.35) has unbounded optimal solutions.
3. If $(\bar{x}_1, \bar{x}_2 \ldots \bar{x}_n)$ is optimal for (3.16) then $(\bar{x}_1 \ldots \bar{x}_n, \bar{a}_1 \ldots \bar{a}_n)$ is optimal for (3.35), where $\bar{a}_j$ is chosen so that $\bar{a}_j \geq A_j\bar{x}_j$, $\sum_{j=1}^{n} \bar{a}_j = a$. If $(\bar{x}_1 \ldots \bar{x}_n, \bar{a}_1 \ldots \bar{a}_n)$ is optimal for (3.35), then $(\bar{x}_1 \ldots \bar{x}_n)$ is optimal for (3.16).

Problem (3.35) is of a kind suitable for the Benders algorithm. It has two groups of variables: $(x_1 \ldots x_n)$ and $(a_1 \ldots a_n)$. The a variables will be taken to

correspond to y in (3.23). A small difference between (3.35) and (3.23) is that the set of $(a_1 \dots a_n)$ satisfying $\sum a_j = a$ [corresponding to Y in (3.23)] is not bounded. This has consequences that are discussed further in section 6.3.3. For given $(a_1 \dots a_n)$, (3.35) decomposes into n separate problems. The jth may be written as

$$\begin{aligned} &\text{Maximize} && c_j x_j \\ &\text{s.t.:} && A_j x_j \leq a_j, \\ & && B_j x_j \leq b_j, \\ & && x_j \geq 0. \end{aligned} \tag{3.36}$$

Precisely as in the discussion of the Benders algorithm in section 3.5.1, a vector a_j is feasible for (3.36) if and only if

$$(\tilde{u}_j^1, \tilde{u}_j^2)^r (a_j, b_j) \geq 0$$

for all extreme rays $(\tilde{u}_j^1, \tilde{u}_j^2)^r$ $(r = 1 \dots R(j))$ of the cone $\{(u_j^1, u_j^2) \mid u_j^1 A_j + u_j^2 B_j \geq 0, (u_j^1, u_j^2) \geq 0\}$. The set $\mathscr{A}_j$ of a_j such that (3.36) has a feasible solution may hence be written:

$$\mathscr{A}_j = \{a_j \mid (\tilde{u}_j^1, \tilde{u}_j^2)^r (a_j, b_j) \geq 0,\ r = 1 \dots R(j)\}.$$

Now let $(u_j^1, u_j^2)^p$ $(p = 1 \dots P(j))$ be the set of extreme points of $\{(u_j^1, u_j^2) \mid u_j^1 A_j + u_j^2 B_j \geq c_j, (u_j^1, u_j^2) \geq 0\}$. Consider any $a_j \in \mathscr{A}_j$. By duality relations in linear programming, it must hold that

$$\begin{aligned} &\max \{c_j x_j \mid A_j x_j \leq a_j,\ B_j x_j \leq b_j,\ x_j \geq 0\} \\ &\quad = \min_{p = 1 \dots P(j)} \{(u_j^1, u_j^2)^p (a_j, b_j)\} \\ &\quad \leq (u_j^1, u_j^2)^p (a_j, b_j) \qquad \text{for } p = 1 \dots P(j). \end{aligned}$$

It is seen that (3.35) may be rewritten in the following equivalent manner, corresponding to problem (3.28) in section 3.5.1:

$$\begin{aligned} &\text{Maximize} && z_1 + z_2 + \cdots + z_n \\ &\text{s.t.:} \\ &z_j \leq (u_j^1, u_j^2)^p (a_j, b_j) && p = 1 \dots P(j), \quad j = 1 \dots n, \\ &a_j \in \mathscr{A}_j && j = 1 \dots n, \\ &a_1 + a_2 + \cdots + a_n = a. \end{aligned} \tag{3.37}$$

From this formulation, it is clear how the Benders algorithm can be implemented: One starts each iteration with a simplified version of (3.37), where, for each j, only a subset of the restrictions $z_j \leq (u_j^1, u_j^2)^p (a_j, b_j)$ and $(\tilde{u}_j^1, \tilde{u}_j^2)^r (a_j, b_j) \geq 0$ are known. That simplified version of (3.37) is, of course,

the supremal subproblem. This results in some solution $(\bar{z}_1, \bar{z}_2 \ldots \bar{z}_n; \bar{a}_1, \bar{a}_2 \ldots \bar{a}_n)$.* For each $j = 1 \ldots n$, one then solves problem (3.36), given $a_j = \bar{a}_j$ [or one solves the dual of (3.36)]. If the dual of (3.36) has solution values going to $-\infty$ along the extreme ray $(\tilde{u}_j^1, \tilde{u}_j^2)'$, the corresponding constraint $(\tilde{u}_j^1, \tilde{u}_j^2)'(a_j, b_j) \geq 0$ is added to the supremal subproblem. If the dual of (3.36) has a finite optimal solution—the extreme point $(u_j^1, u_j^2)^p$, with solution value $(u_j^1, u_j^2)^p(\bar{a}_j, b_j) < \bar{z}_j$—the corresponding constraint $(u_j^1, u_j^2)^p(a_j, b_j) \geq z_j$ is added to the supremal subproblem. If (3.36) (or its dual) has a finite optimal solution value $(u_j^1, u_j^2)^p(\bar{a}_j, b_j) \geq \bar{z}_j$, then no constraint is added to the supremal subproblem for that index j.† If each problem (3.36) (or its dual) has a finite optimal solution value $(u_j^1, u_j^2)^p(\bar{a}_j, b_j)$ and if $(u_j^1, u_j^2)^p(\bar{a}_j, b_j) \geq \bar{z}_j$ for all $j = 1 \ldots n$, then the process stops. An optimal solution to (3.37) has been obtained.

This method of solving LP problems with block-angular structures has been discussed by several authors; see, e.g., Freeland and Baker (1975), Geoffrion (1970a), or ten Kate (1972). The algorithm is sometimes referred to as the ten Kate algorithm, but it is really a direct extension of the Benders algorithm.

The Benders algorithm has a very clear two-level interpretation in the context of the overall problem (3.35). As was mentioned in section 3.3.6, a block-angular LP problem may be interpreted as a planning problem in a divisionally organized corporation, where the divisions jointly utilize certain scarce resources. This interdependence is expressed by the vector inequality $A_1x_1 + A_2x_2 + \cdots + A_nx_n \leq a$. Under the Benders algorithm, there is a supremal subproblem that involves allocations of the joint resources between divisions. These allocations are given by the vectors $a_1 \ldots a_n$. There are n infimal subproblems, each of the type (3.36). The infimal subproblem of division j consists of finding an optimal production program for that division, taking into account the given allocation vector a_j. Denote the optimal solution value for (3.36) by $v_j(a_j)$ (which may be set to $-\infty$ if $a_j \notin \mathscr{A}_j$). For some $\bar{a}_j$, suppose (3.36) has a finite optimal solution. Let $(\bar{u}_j^1, \bar{u}_j^2)$ be an optimal dual solution. For some other $a_j \in \mathscr{A}_j$,

$$\begin{aligned} v_j(a_j) &= \min \{(u_j^1, u_j^2)(a_j, b_j) \mid u_j^1 A_j + u_j^2 B_j \geq c_j,\ (u_j^1, u_j^2) \geq 0\} \\ &\leq \bar{u}_j^1 a_j + \bar{u}_j^2 b_j. \end{aligned}$$

* The possibility of unbounded $\bar{a}_1 \ldots \bar{a}_n$ may be eliminated by imposing on the supremal subproblem restrictions $a_j \leq M$, where M is a vector with "large" components. See also section 6.3.3.

† For completeness: If the dual of (3.36) is infeasible, a special investigation is necessary. This is the situation discussed in section 3.5.2.

It also holds that $v_j(\bar{a}_j) = \bar{u}_j^1 \bar{a}_j + \bar{u}_j^2 b_j$. That is, one has the two relations

$$0 = v_j(\bar{a}_j) - \bar{u}_j^1 \bar{a}_j - \bar{u}_j^2 b_j,$$

$$v_j(a_j) \leq \bar{u}_j^1 a_j + \bar{u}_j^2 b_j.$$

Adding these two relations, one obtains $v_j(a_j) \leq v_j(\bar{a}_j) + \bar{u}_j^1 (a_j - \bar{a}_j)$. The information sent back from subproblem (3.36) includes $\bar{u}_j^1$ if $\bar{a}_j \in \mathcal{A}_j$. From this discussion, it is seen that $\bar{u}_j^1$ may be interpreted as a piece of information on how the optimal solution value of (3.36) would change in response to small changes in $\bar{a}_j$. Or, even more specifically, $\bar{u}_j^1$ may be interpreted as a vector of maximal prices that the division would be willing to pay for incremental amounts of the joint resources. Hence, one could say that the supremal subproblem sends information about quantities to the infimal subproblems under the Benders algorithm. The infimal subproblems respond with price information.

The Benders algorithm, as applied to the block-angular LP problem (3.16), is an instance of the idealized multilevel approach discussed in section 2.1. A two-level subproblem hierarchy is constructed. There are n infimal subproblems of the type (3.36). The jth infimal subproblem is parameterized by the vector a_j. The supremal subproblem may be described as one of finding an optimal allocation, or partitioning, of the right-hand side a of the coupling constraints among the infimal subproblems. The adjustment phase is as described above: i.e., an iterative exchange of quantity and price information. In the execution phase, the resulting solution to the overall problem (3.16) is recovered from the infimal subproblems (3.36).

3.5.5 ON THE RELATION BETWEEN THE BENDERS AND DANTZIG–WOLFE ALGORITHMS

From what has been said so far about the Dantzig–Wolfe and Benders algorithms, one can detect certain relationships between the two. The supremal subproblem of the Dantzig–Wolfe algorithm is successively extended with additional columns, whereas the supremal subproblem under the Benders algorithm is extended with additional rows (restrictions) as the algorithm proceeds. Also, the information flows under the Dantzig–Wolfe algorithm may be briefly described as follows: the supremal subproblem sends prices to the infimal subproblems; these respond with quantities. The Benders algorithm involves precisely the opposite flows, as was pointed out at the end of section 3.5.4. In a formal sense, the two algorithms may indeed be regarded as dual ones. This holds only when they are applied to linear problems like (3.35), however. The reason is that the Dantzig–Wolfe algorithm, as has been discussed here, has not been defined for mixed integer programming problems, among others.

Consider, therefore, problem (3.35) again. The dual of (3.35) may be written as

$$\begin{aligned}
&\text{Minimize} \quad u_1^2 b_1 + u_2^2 b_2 + \cdots + u_n^2 b_n + \pi a \\
&\text{s.t.:} \qquad -u_j^1 + \pi = 0 \qquad (j = 1 \ldots n), \quad \text{(coupling constraints)} \\
&\qquad u_j^1 A_j + u_j^2 B_j \geq c_j \qquad (j = 1 \ldots n), \\
&\qquad (u_j^1, u_j^2) \geq 0 \qquad (j = 1 \ldots n).
\end{aligned} \tag{3.38}$$

Suppose now one applies Dantzig–Wolfe decomposition problem (3.38). The Dantzig–Wolfe extremal problem is then

$$\begin{aligned}
&\text{Minimize} \quad \sum_{j=1}^{n} \left\{ \sum_{p=1}^{P(j)} \lambda_j^p (u_j^2)^p b_j + \sum_{r=1}^{R(j)} \delta_j^r (\tilde{u}_j^2)^r b_j \right\} + \pi a \\
&\text{s.t.:} \quad -\sum_{p=1}^{P(j)} \lambda_j^p (u_j^1)^p - \sum_{r=1}^{R(j)} \delta_j^r (\tilde{u}_j^1)^r + \pi = 0 \qquad (j = 1 \ldots n), \\
&\qquad\qquad \sum_{p=1}^{P(j)} \lambda_j^p = 1 \qquad (j = 1 \ldots n), \\
&\qquad\qquad \lambda_j^p, \delta_j^r \geq 0.
\end{aligned} \tag{3.39}$$

If one now takes the dual of (3.39), denoting the dual variables of the first n vector equalities by a_j and of the n convexity rows by z_j, then one obtains problem (3.37), i.e., the equivalent full master problem of the Benders algorithm. From this, it is not difficult to see that applying the Benders algorithm to (3.35) is entirely equivalent to dualizing (3.35) and then applying Dantzig–Wolfe to the dual. The information flows from the supremal subproblem to the infimal ones and from the infimal subproblems to the supremal one will be exactly the same in both cases. The optimality tests in the two cases also involve the same condition.

3.6 THE KORNAI–LIPTAK DECOMPOSITION ALGORITHM

The decomposition algorithm that we will discuss in this section arose in the context of economic planning and was presented in Kornai and Liptak (1965). It turns out that this algorithm is a simplified version of the Benders algorithm when applied to the block-angular problem (3.16). These simplifications are a consequence of a few additional assumptions. In the following discussion, we assume that two conditions are met:

- The sets $\mathscr{A}_j$ (defined in section 3.5.4) can be completely specified in advance as $\mathscr{A}_j = (E^m)^+$, i.e., the set of nonnegative m vectors.
- The sets $\{x_j \mid B_j x_j \leq b_j,\ x_j \geq 0\}$ are bounded.

With respect to the first condition, it follows easily that if $0 \in \{x_j \mid B_j x_j \leq b_j, x_j \geq 0\}$ and $A_j \geq 0$ for $j = 1 \ldots n$, this condition is met.

We can now specify the Kornai–Liptak algorithm for solving problem (3.35) satisfying the above conditions.

Iteration 1 (initialization)

Step 1. Select any $a_j(1)$, $j = 1 \ldots n$, such that $a_j(1) \geq 0$ and $\sum a_j(1) = a$.

Step 2. Set $a_j\langle 1\rangle = a_j(1)$ for $j = 1 \ldots n$.

Step 3. Solve the subproblem, for $j = 1 \ldots n$.

$$\begin{aligned} &\text{Maximize} && c_j x_j \\ &\text{s.t.:} && A_j x_j \leq a_j\langle 1\rangle, \\ & && B_j x_j \leq b_j, \\ & && x_j \geq 0. \end{aligned}$$

Let $u_j^1(1)$ and $u_j^2(1)$ be optimal dual multiplier vectors associated with $A_j x_j \leq a_j\langle 1\rangle$ and $B_j x_j \leq b_j$. A lower bound on the optimal solution value of (3.35) is given by $\sum_{j=1}^{n} (u_j^1(1) a_j\langle 1\rangle + u_j^2(1) b_j)$.

Step 4. Set $u_j^1\langle 1\rangle = u_j^1(1)$, $u_j^2\langle 1\rangle = u_j^2(1)$, $j = 1 \ldots n$.

Iteration t

Step 1. Solve the LP problem:

$$\begin{aligned} &\text{Maximize} && \sum_{j=1}^{n} u_j^1\langle t-1\rangle a_j \\ &\text{s.t.:} && \sum_{j=1}^{n} a_j = a, \\ & && a_j \geq 0, \qquad j = 1 \ldots n. \end{aligned}$$

Let $a_j(t)$, $j = 1 \ldots n$, be an optimal solution. The quantity $\sum_{j=1}^{n} (u_j^1\langle t-1\rangle a_j(t) + u_j^2\langle t-1\rangle b_j)$ is an upper bound on the optimal solution value of (3.35). Note that this problem is solved by allocating *all* of the ith common resource to that division j that has the highest objective function coefficient for that particular resource.

Step 2. Set

$$\begin{aligned} a_j\langle t\rangle &= \frac{1}{t} \sum_{s=1}^{t} a_j(s) \\ &= \frac{t-1}{t} a_j\langle t-1\rangle + \frac{1}{t} a_j(t), \qquad j = 1 \ldots n. \end{aligned}$$

Step 3. Solve the subproblems, for $j = 1 \ldots n$:

$$\begin{aligned} &\text{Maximize} && c_j x_j \\ &\text{s.t.:} && A_j x_j \leq a_j\langle t\rangle, \\ & && B_j x_j \leq b_j, \\ & && x_j \geq 0. \end{aligned}$$

Let $(u_j^1(t), u_j^2(t))$ be an optimal dual multiplier vector. Now $\sum_{j=1}^{n} (u_j^1(t) a_j\langle t\rangle + u_j^2(t) b_j)$ gives a lower bound on the optimal solution value of (3.35).

Step 4. Set

$$\begin{aligned} u_j^i\langle t\rangle &= \frac{1}{t} \sum_{s=1}^{t} u_j^i(s) \\ &= \frac{t-1}{t} u_j^i\langle t-1\rangle + \frac{1}{t} u_j^i(t), \qquad j = 1 \ldots n,\ i = 1; 2. \end{aligned}$$

Go to iteration $t+1$.

This is the adjustment phase of the Kornai–Liptak algorithm. In Step 1 of each iteration, the supremal subproblem is solved, in Step 3, the infimal subproblems. Steps 2 and 4 deal with the iterative information exchange between supremal and infimal subproblems. It can be demonstrated that the sequence of allocation vectors $(a_1\langle t\rangle \ldots a_n\langle t\rangle)$ will converge toward optimal allocation vectors as t goes to infinity. Finite convergence does not occur except in very special cases. Convergence follows from certain game-theoretic results on the solution of two-person zero-sum games by fictitious play (see Gale 1960, p. 246, for a discussion of fictitious play). However, since finite convergence does not occur, the adjustment phase must usually be terminated before optimal allocation vectors have been identified. In the execution phase, a feasible solution to (3.16) is obtained from the infimal subproblems.

Apparently, the supremal subproblem in Step 1 of each iteration is a greatly simplified version of (3.37). In (3.37), there may be a very large number of restrictions of the type $z_j \leq (u_j^1, u_j^2)^p (a_j, b_j)$ for each index j. Storing all those restrictions may not be feasible because of limited computer memory capacity. The Kornai–Liptak algorithm was developed precisely to economize on computer memory capacity.

3.7 LAGRANGEAN DECOMPOSITION IN NONLINEAR PROGRAMMING

The discussion in the previous sections focused on linear models and their immediate extensions; here, however, we will present a two-level method,

often referred to as Lagrangean decomposition, for nonlinear mathematical programming problems with "separable" objective functions and constraint sets. The basic ideas underlying the approach were already present in early work of Uzawa (1958). In Lasdon (1968) and Geoffrion (1971), duality theory is employed to derive the method. The reader may also find an extensive discussion in Lasdon (1970, Chapter 8).

3.7.1 LAGRANGEAN DECOMPOSITION FOR SEPARABLE MATHEMATICAL PROGRAMMING PROBLEMS

Consider the following mathematical programming problem, the overall problem of this section:

$$\begin{aligned} &\text{Minimize} && \sum_{j=1}^{n} f_j(x_j) \\ &\text{s.t.:} && \sum_{j=1}^{n} g_j(x_j) \leq a, \\ & && x_j \in X_j, \qquad j = 1 \ldots n. \end{aligned} \tag{3.40}$$

The following assumptions are imposed on problem (3.40): The variables x_j, $j = 1 \ldots n$, are in Euclidean n_j space, E^{n_j}, and are restricted to convex and compact subsets $X_j \subset E^{n_j}$. The real-valued functions f_j and the m-dimensional vector-valued functions g_j are continuous on X_j. Moreover, each f_j is strictly convex on X_j, and each component of g_j is convex. Of course, $a \in E^m$. Denoting $x = (x_1 \ldots x_n)$ and $X = X_1 \times X_2 \times \cdots \times X_n$, we also assume the existence of an $x' \in X$ such that $\sum g_j(x_j') < a$. This last assumption is the well-known Slater constraint qualification and guarantees (together with the other conditions) not only the existence of a unique optimal solution $\bar{x} = (\bar{x}_1, \bar{x}_2 \ldots \bar{x}_n)$ to (3.40), but also a nonempty set of optimal dual multiplier vectors associated with the constraint $\sum g_j(x_j) \leq a$. Let that set of optimal dual multipliers be denoted $\bar{U}$. $\bar{U}$ could have more than one member. For $\bar{u} \in \bar{U}$ it holds that:

1. $(\bar{x}_1, \bar{x}_2 \ldots \bar{x}_n)$ minimizes $\sum_{j=1}^{n} f_j(x_j) + \bar{u}[\sum_{j=1}^{n} g_j(x_j) - a]$ over X
2. $\bar{u} \geq 0$
3. $\sum_{j=1}^{n} g_j(\bar{x}_j) \leq a$
4. $\bar{u}\left(\sum_{j=1}^{n} g_j(\bar{x}_j) - a\right) = 0$

(1)–(4) are necessary and sufficient optimality conditions for an optimal solution $\bar{x}$ to (3.40).

Consider now the following two-level representation of the overall problem (3.40). There are n infimal subproblems, written (for $j = 1 \ldots n$) as:

$$\begin{aligned} &\text{Minimize} && f_j(x_j) + u g_j(x_j) \\ &\text{s.t.:} && x_j \in X_j. \end{aligned} \tag{3.41}$$

Let $x_j(u)$ denote the (unique) optimal solution to (3.41). Evidently, (3.41) is parameterized by the m vector u, which may be interpreted as a price vector. The supremal subproblem may be stated: Find u' such that $(x_1(u'), x_2(u') \ldots x_n(u')) = \bar{x}$. Under the assumptions made, the set of solutions to the supremal subproblem is precisely the set $\bar{U}$. This two-level subproblem hierarchy is hence coordinable relative to the original problem (3.40).

A "price-adjustment procedure" can now be employed for finding a solution $\bar{u}$ to the supremal subproblem. It may be outlined as follows:

Iteration 1 (initialization)

Step 1. Pick any $u' \geq 0$.

Step 2. For $j = 1 \ldots n$, solve (3.41) using $u = u'$.

Iteration t

Step 1. A new trial price vector u^t is determined componentwise:

$$u_i^t = \max\left\{0,\ u_i^{t-1} + \alpha^t\left(\sum_{j=1}^{n} g_{ij}(x_j(u^{t-1})) - a_i\right)\right\},$$

where i indexes the components of $g_j(x_j)$, a, and u^t. The adjustment constant α^t is chosen positive.

Step 2. For $j = 1 \ldots n$, solve (3.41) using $u = u^t$.

This may indeed be interpreted as a price-adjustment process. In Step 1 of each iteration (except iteration 1), the price associated with each coupling constraint is increased or decreased, depending on whether there is an excess demand or supply relating to the right-hand side of the relevant constraint. One may demonstrate (Uzawa 1958) that u^t converges to $\bar{u} \in \bar{U}$ as $t \to \infty$, provided $\alpha^t = \alpha > 0$ but sufficiently small. Moreover, one can show that as u^t converges to $\bar{u}$, $x_j(u^t)$ converges to $x_j(\bar{u}) = \bar{x}_j$ (Falk 1967, p. 151).

The multilevel solution method described here is another instance of the idealized multilevel approach of section 2.1. The information exchange between supremal and infimal subproblems in the adjustment phase is described above; in the execution phase, a solution to the overall problem (3.40) is recovered from the infimal subproblems. The name Lagrangean decomposition derives from the fact that the function $\sum_{j=1}^{n} f_j(x_j) + u(\sum_{j=1}^{n} g_j(x_j) - a)$ is called *the Lagrangean.* Also, the dual multipliers u are sometimes called *Lagrange multipliers.*

Lagrangean decomposition, as presented here, presupposes that the objective function of the overall problem is strictly convex. This rules out linear programming problems. Indeed, the infimal subproblems may have multiple optimal solutions in the linear case, so the adjustment in Step 1 of each iteration may not be well defined. Also, this subproblem hierarchy is not coordinable

relative to an overall problem of the linear type, a fact already mentioned in section 2.1.2. Nevertheless, this method may be used for linear overall problems after suitable nonlinear perturbation (Jennergren 1973; Poliak and Tret'iakov 1972).

3.7.2 DUALITY THEORY AND LAGRANGEAN DECOMPOSITION

Consider $u \geq 0$. As before, the Lagrangean function is

$$\begin{aligned} L(x, u) &= \sum_{j=1}^{n} f_j(x_j) + u\left(\sum_{j=1}^{n} g_j(x_j) - a\right) \\ &= \sum_{j=1}^{n} (f_j(x_j) + ug_j(x_j)) - ua \\ &\equiv \sum_{j=1}^{n} L_j(x_j, u) - ua. \end{aligned}$$

Now, for any given $u \geq 0$,

$$\min_{x \in X} L(x, u) = \sum_{j=1}^{n} \min_{x_j \in X_j} L_j(x_j, u) - ua,$$

and we already recognize the subproblems (3.41). We now define the *dual function* as

$$h(u) = \min_{x \in X} L(x, u).$$

Since f_j and g_j are continuous and X_j is compact, the domain of definition of the dual function is simply E^m. The dual program associated with (3.40) is defined as

$$\begin{aligned} &\text{Maximize} \quad h(u) \\ &\text{s.t.:} \qquad\quad u \geq 0. \end{aligned} \tag{3.42}$$

Under the assumptions made earlier, $h(u)$ is concave and everywhere differentiable (Lasdon 1970, pp. 419–428). Also, one can demonstrate the following proposition: If $\bar{u}$ is an optimal solution to (3.42), then $x(\bar{u})$ such that $L(\bar{x}, \bar{u}) = \min_{x \in X} L(x, \bar{u})$ is an optimal solution to (3.40). Hence, solving (3.42) gives a solution to (3.40) by means of the infimal subproblems (3.41).

Indeed, the price-adjustment procedure attempts precisely to solve the dual problem (3.42). Suppose the vector u^t is given at some iteration t. Solving the infimal subproblems gives $x_j(u^t)$, $j = 1 \ldots n$, and one obtains $h(u^t) = \sum_{j=1}^{n} L_j(x_j(u^t), u^t) - u^t a$. If one now wants to choose a new price vector, u^{t+1}, which maximizes the initial rate of improvement of the dual function, one must

evaluate the gradient of h at u^t. One can demonstrate that

$$\nabla h(u)|_{u=u^t} = \sum_{j=1}^{n} g_j(x_j(u^t)) - a.$$

Hence the price-adjustment procedure attempts to solve the dual by means of a steepest-ascent algorithm.

3.8 HEURISTIC METHODS

All the methods discussed above converge on an optimal solution to the overall problem with which one began. Quite often, one will find problem structures that do not satisfy certain assumptions required by these methods. There may, for instance, be nonconvexities or integer variables in problems otherwise suited for column-generation techniques. Then heuristic methods may be applied. Also, there are other problem situations where a theoretically convergent method could in principle be applied, but a heuristic one is nevertheless used, since that is a practical way of quickly obtaining a good solution to the overall problem. In this book, we will see instances of problem situations where heuristic multilevel methods are used. In particular, the three case studies in Chapter 5 all involve heuristic methods. The details of these methods are rather problem-oriented and will be discussed in the relevant parts of the subsequent chapters.

3.9 MULTILEVEL CONTROL THEORY: A BRIEF SURVEY

The methods presented in the preceding sections of this chapter are all applicable to mathematical programming problems of the general type

$$\begin{aligned} &\text{Maximize} \quad f(x) \\ &\text{s.t.:} \qquad g(x) \leq 0. \end{aligned} \tag{3.43}$$

with appropriate conditions imposed on them. As it turns out, many control problems can be cast in this format, which implies that the techniques described earlier in this chapter may be regarded as belonging to the realm of multilevel control theory. However, the modern theory of multilevel control pertains to a larger class of optimization problems than (3.43). Our object in this section is to give a brief survey of multilevel control theory and to identify classes of control problems that can be solved by techniques of the type presented earlier in this chapter.

The approach and symbolism of this section will be that of "systems science" rather than "mathematical programming." In view of the expository nature of

the discussion, we will not strive for rigor, in order to avoid various mathematical issues.

3.9.1 STATIC MULTILEVEL CONTROL PROBLEMS

A fairly general class of static multilevel control problems with n subsystems, relating to, for example, complex chemical processes in the steady state, may be written (Mahmoud 1977):

$$\begin{aligned} &\text{Minimize} && \sum_{j=1}^{n} f_j(x_j, m_j) \\ &\text{s.t.:} && y_j = S_j(x_j, m_j), \\ & && z_j = T_j(x_j, m_j), \\ & && x_j = \sum_{j=1}^{n} h_{ji} z_i, \\ & && g_j(m_j, x_j, y_j, z_j) \leq 0 \qquad (j = 1 \ldots n). \end{aligned} \tag{3.44}$$

To clarify (3.44), we make reference to Figure 3.7, representing a typical subsystem j. The pair of vectors (x_j, m_j) represents the input. x_j stands for the input coming from other subsystems, and m_j is the control vector; the output is given by the pair (y_j, z_j), y_j being the final output and z_j the "output coupling" vector. The functions S_j and T_j, then, determine the input–output relation of subsystem j. The equalities $x_j = \sum h_{ji} z_i$ represent the coupling constraints, where the coefficients h_{ji} are output–input transformation coefficients. Finally, the (vector-valued) functions $g_j(m_j, x_j, y_j, z_j)$ determine a feasible region for each subsystem. It is an obvious observation that if enough conditions are imposed on the functions f_j, S_j, T_j, and g_j, (3.44) can be solved by appropriate multilevel techniques of mathematical programming. However, the presence of nonlinear equality constraints sets (3.44) apart from more conventional mathematical programming problems.

In the systems science literature, three main approaches are suggested for solving problems of type (3.44): the parametric method, the dual coordination method, and the penalty function method (see Kulikowski *et al.* 1975). The

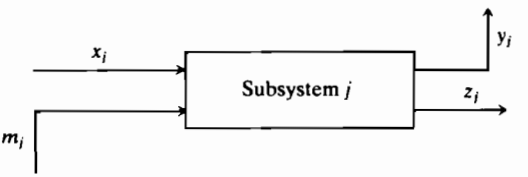


FIGURE 3.7 Representation of a subsystem.

parametric method corresponds to primal decomposition methods as presented in Geoffrion (1970a). (The Benders algorithm, when applied to a block-angular LP problem, is a primal decomposition method.) The dual coordination method is precisely the method discussed in section 3.7 (Lagrangean decomposition). For discussions of the penalty function method in this context, see Tatjewski (1975) or Findeisen *et al.* (in press). Schoeffler (1971) gives a more extensive discussion of static multilevel control theory.

3.9.2 DYNAMIC OPEN-LOOP MULTILEVEL CONTROL

We now focus on dynamic systems. The restriction to open-loop control structures, in contrast to closed-loop or on-line control, means that a model is formulated for which a policy is to be determined for the entire planning period *without* the use of feedback information since the system is not yet operating. As Findeisen points out (1976, p. 3), this is the case with planning (or scheduling), and the goodness of the policy is dependent only on the accuracy of the model. We first consider discrete-time systems and then turn to continuous-time ones. The exposition will follow Singh *et al.* (1975) and Smith and Sage (1973). (See also Singh 1977.)

Solution techniques for multilevel discrete-time control problems correspond to well-known optimization methods, including methods discussed earlier in this chapter. Unlike the static case, however, dual coordination or Lagrangean methods are really the only significant ones for wide classes of problems, as pointed out by Singh *et al.* (1975). We follow their paper in discussing only dual coordination methods. One version of the discrete-time control problem with separable objective function may be formulated as follows:

$$\text{Minimize} \quad \sum_{j=1}^{n} \left(\sum_{t=0}^{T-1} f_{jt}(x_j(t), m_j(t)) + f_{jT}(x_j(T)) \right) \tag{3.45}$$

s.t.:

$$x_j(0) = \underline{x}_j,$$

$$x_j(t+1) = k_{jt}(x_j(t), m_j(t)) \qquad (j = 1 \ldots n;\ t = 0, 1 \ldots T-1), \tag{3.46}$$

$$\sum_{j=1}^{n} h_{jt}(x_j(t), m_j(t)) = 0 \qquad (t = 0, 1 \ldots T-1), \tag{3.47}$$

$$g_{jt}(x_j(t), m_j(t)) \leq 0 \qquad (j = 1 \ldots n;\ t = 0, 1 \ldots T-1), \tag{3.48}$$

$$g_{jT}(x_j(T)) \leq 0 \qquad (j = 1 \ldots n). \tag{3.49}$$

The interpretation of (3.45) is straightforward: each function f_{jt} ($t = 0, 1 \ldots T-1$) measures the cost in subsystem j in period t as a function of the state vector $x_j(t)$ and the control $m_j(t)$; the cost in the last period depends on the

terminal value of the state vector. The system dynamics are given in (3.46) by first-order nonlinear difference equations. The coupling constraints are given by (3.47), whereas the inequalities (3.48) and (3.49) determine the feasible regions of the state and control vectors locally. Without further assumptions, this problem may be very difficult to solve, but if one imposes enough conditions on it, one recognizes it as a decomposable convex programming problem with the $m_j(t)$ as decision variables. One may adapt the Lagrangean decomposition technique (see section 3.7) to the discrete-time problem described above. If we associate a Lagrange multiplier vector $\lambda(t)$ with (3.47), we see immediately that applying Lagrangean decomposition results in the following subproblems ($j = 1 \ldots n$):

$$\begin{aligned}
&\text{Minimize} && \sum_{t=0}^{T-1} \{f_{jt}(x_j(t), m_j(t)) + \lambda(t) h_{jt}(x_j(t), m_j(t))\} + f_{jT}(x_j(T)) \\
&\text{s.t.:} && x_j(0) = \underline{x}_j, \\
& && x_j(t+1) = k_{jt}(x_j(t), m_j(t)) \qquad (t = 0, 1 \ldots T-1), \\
& && g_{jt}(x_j(t), m_j(t)) \leq 0 \qquad (t = 0, 1 \ldots T-1), \\
& && g_{jT}(x_j T) \leq 0.
\end{aligned} \tag{3.50}$$

One may exploit the idea of Lagrangean decomposition further to obtain a three-level method for solving the discrete-time problem, as is also discussed in Singh *et al.* (1975). This is done by associating with the constraints $x_j(t+1) = k_{jt}(x_j(t), m_j(t))$ of (3.50) another Lagrange multiplier vector $\mu_j(t)$. Subproblem (3.50) then decomposes into T smaller subproblems.

This discussion was only intended to give the reader an impression of the importance of Lagrangean decomposition in discrete-time control problems. Singh *et al.* (1975) consider Lagrangean decomposition of discrete-time problems with time delay—systems where higher-order difference equations are allowed for. They discuss briefly the application of this method to the control of urban road traffic signals. Another application of Lagrangean decomposition, to a dynamic production-planning problem, is discussed in Drew (1975).

By now it should be clear that the difference between the methods presented in sections 3.2–3.7 and those of multilevel systems control of static or discrete-time systems is at least partly one of terminology. This is not the case, however, for continuous-time control problems. We will now indicate how a two-level representation of a continuous-time overall problem can be constructed. Consider the following overall problem:

$$\begin{aligned}
&\text{Minimize} && \Phi(x(T)) + \int_0^T \phi(x(t), m(t), t)\, dt \\
&\text{s.t.:} && \dot{x} = f(x(t), m(t), t), \\
& && x(0) = \underline{x}.
\end{aligned} \tag{3.51}$$

Problem (3.51) is a standard control problem. If the objective functional is assumed to be separable, i.e.,

$$\phi = \sum_i \phi_i(x_i, m_i, t),$$

$$\Phi = \sum_i \Phi_i(x_i(T)),$$

and if the constraints $\dot{x} = f(x(t), m(t), t)$ can be decomposed, one can rewrite (3.51) as follows:

$$\begin{aligned} \text{Minimize} \quad & \sum_{j=1}^{n} \left\{ \Phi_j(x_j(T)) + \int_0^T \phi_j(x_j(t), m_j(t), t)\, dt \right\} \\ \text{s.t.:} \quad & \dot{x}_j = f_j(x_j(t), v_j(t), m_j(t), t) \qquad (j = 1 \ldots n), \\ & x_j(0) = \underline{x}_j \qquad (j = 1 \ldots n), \\ & v_j(t) = \sum_{\substack{i=1 \\ i \neq j}}^{n} g_{ji}(x_i, m_i) \qquad (j = 1 \ldots n). \end{aligned} \tag{3.52}$$

To assume coupling constraints as in (3.52) means some loss of generality, but it is quite essential under the present approach. The variables $v_j(t)$ are referred to as the coordinating variables. Observe that the functions f_j in (3.52) are not merely components of f in problem (3.51); this is the case because of the appearance of the coordinating variables.

Now suppose one uses multiplier time functions $\mu_j(t)$ $(j = 1 \ldots n)$ to eliminate the coupling constraints from (3.52). One obtains yet another problem formulation:

$$\begin{aligned} \text{Minimize} \quad & \sum_{j=1}^{n} \left\{ \Phi_j + \int_0^T \left[\phi_j + \mu_j \left(v_j - \sum_{\substack{i=1 \\ i \neq j}}^{n} g_{ji}(x_i, m_i) \right) \right] dt \right\} \\ & = \sum_{j=1}^{n} \left\{ \Phi_j + \int_0^T \left[\phi_j + \mu_j v_j - \sum_{\substack{i=1 \\ i \neq j}}^{n} \mu_i g_{ij}(x_j, m_j) \right] dt \right\} \\ \text{s.t.:} \quad & \dot{x}_j = f_j(x_j, v_j, m_j, t) \qquad (j = 1 \ldots n), \\ & x_j(0) = \underline{x}_j \qquad (j = 1 \ldots n). \end{aligned}$$

But this problem formulation decomposes directly into n subproblems, for $j = 1 \ldots n$:

$$\begin{aligned} \text{Minimize} \quad & \Phi_j + \int_0^T \left[\phi_j + \mu_j v_j - \sum_{i \neq j} \mu_i g_{ij}(x_j, m_j) \right] dt \\ \text{s.t.:} \quad & \dot{x}_j = f_j(x_j, v_j, m_j, t), \\ & x_j(0) = \underline{x}_j. \end{aligned} \tag{3.53}$$

The problems (3.53) are the infimal subproblems. Under certain conditions there exist $\mu^* = (\mu_1^* \ldots \mu_n^*)$ such that a collection of optimal solutions to the infimal subproblems constitutes an optimal solution to the original problem (3.51) (Smith and Sage 1973). The supremal subproblem may then be stated as that of finding μ^*.

The question then becomes: What solution methods can be used to solve (3.51) in a multilevel fashion? Or, equivalently, what methods can be used to solve the supremal subproblem and find μ^*? This matter will not be pursued here, but the reader is referred to Pearson (1971) and Smith and Sage (1973) for discussions of multilevel solution methods for continuous-time control problems. This volume is not intended to survey such methods, since one may argue that continuous-time models probably find fewer applications in economic planning than in engineering situations. Different types of applications are discussed in Mahmoud (1977).

3.9.3 ON-LINE CONTROL MODELS

In section 3.9.2, we were concerned with open-loop control problems. If, however, one is faced with the task of developing control policies for operating systems (usually with disturbances), things become more complex. With respect to multilevel methods for on-line control, existing literature is scarce, as Mahmoud (1977, p. 134) remarks. However, work in the area is currently being undertaken by Findeisen and his associates (Findeisen 1978; Findeisen *et al.* 1978, in press; Findeisen and Malinowski 1978). Some additional references on multilevel on-line control models are Chong and Athans (1975), Singh (1977), and Singh *et al.* (1976). Multilevel on-line control is apparently a research area with great potential, but so far there have not been many real-life applications in economics or management science.

REFERENCES

Adler, I., and A. Ülkücü. 1973. On the number of iterations in Dantzig–Wolfe decomposition algorithm, pp. 181–188. In D. M. Himmelblau (ed.), Decomposition of Large-Scale Problems, Amsterdam: North-Holland.

Almon, C. 1963. Central planning without complete information at the center, pp. 462–466. In G. B. Dantzig, Linear Programming and Extensions, Princeton, New Jersey: Princeton University Press.

Bagrinowski, K. A. 1975. Modelle und Methoden der ökonomischen Kybernetik. (Models and Methods of Economic Cybernetics, in German, translated from Russian.) Berlin: Verlag Die Wirtschaft.

Balinski, M. L., and E. Hellerman (ed.). 1975. Mathematical Programming Study 4: Computational Practice in Mathematical Programming. Amsterdam: North-Holland.

Baumol, W. J., and T. Fabian. 1964. Decomposition, pricing for decentralization, and external economies. Management Science 11: 1–32.

Benders, J. F. 1962. Partitioning procedures for solving mixed-variables programming problems. Numerische Mathematik 4: 238–252.

Bensoussan, A., J. L. Lions, and R. Teman. 1972. Sur les méthodes de décomposition, de décentralisation et de coordination, et applications. (On the methods of decomposition, decentralization, and coordination, with applications, in French.) IRIA Cahiers No. 11: 5–189.

Chong, C. Y., and M. Athans. 1975. On the periodic coordination of linear stochastic systems. In Proceedings of IFAC 6th World Congress. Boston.

Dantzig, G. B. 1963. Linear Programming and Extensions. Princeton, New Jersey: Princeton University Press.

Dantzig, G. B., and R. M. Van Slyke. 1967. Generalized upper bounding techniques. Journal of Computer and System Sciences 1: 213–226.

Dantzig, G. B., and P. Wolfe. 1961. The decomposition algorithm for linear programs. Econometrica 29: 767–778.

Drew, S. 1975. The application of hierarchical control methods to a managerial problem. International Journal of Systems Science 6: 371–395.

Elmaghraby, S. 1970. The theory of networks and management science: Part I. Management Science 17: 1–34.

Falk, J. E. 1967. Lagrange multipliers and nonlinear programming. Journal of Mathematical Analysis and Applications 19: 141–159.

Findeisen, W. 1976. Lectures on Hierarchical Control Systems. Report, Center for Control Sciences. Minneapolis: University of Minnesota.

Findeisen, W. 1978. Hierarchical Control Systems—An Introduction. PP-78-1. Laxenburg, Austria: International Institute for Applied Systems Analysis.

Findeisen, W., M. Brdys, K. Malinowski, P. Tatjewski, and A. Wozniak. 1978. On-line hierarchical control for steady-state systems. IEEE Transactions on Automatic Control 23: 189–209.

Findeisen, W., F. N. Bailey, M. Brdys, K. Malinowski, P. Tatjewski, and A. Wozniak. In press. Control and Coordination in Hierarchical Systems. Chichester, England: Wiley.

Findeisen, W., and K. Malinowski. 1978. Two-level control and coordination for dynamical systems. In IFAC Congress. Helsinki.

Ford, L. R., and D. R. Fulkerson. 1956. Maximal flow through a network. Canadian Journal of Mathematics 8: 399–404.

Ford, L. R., and D. R. Fulkerson. 1958. Suggested computation for maximal multi-commodity network flows. Management Science 5: 97–101.

Ford, L. R., and D. R. Fulkerson. 1962. Flows in Networks. Princeton, New Jersey: Princeton University Press.

Freeland, J. R., and N. R. Baker. 1975. Goal partitioning in a hierarchical organization. Omega 3: 673–688.

Gale, D. 1960. The Theory of Linear Economic Models. New York: McGraw-Hill.

Garfinkel, R. S., and G. L. Nemhauser. 1972. Integer Programming. New York: Wiley.

Geoffrion, A. M. 1970a. Primal resource-directive approaches for optimizing nonlinear decomposable systems. Operations Research 18: 375–403.

Geoffrion, A. M. 1970b. Elements of large-scale mathematical programming. Management Science 16: 652–691.

Geoffrion, A. M. 1971. Duality in nonlinear programming: A simplified applications-oriented development. SIAM Review 13: 1–37.

Hagelschuer, P. B. 1971. Theorie der linearen Dekomposition. (Theory of Linear Decomposition, in German). Berlin: Springer-Verlag.

Jarvis, J. J. 1969. On the equivalence between the node–arc and arc–chain formulations for the multi-commodity maximal flow problem. Naval Research Logistics Quarterly 16: 525–529.

Jennergren, L. P. 1973. A price schedules decomposition algorithm for linear programming problems. Econometrica 41: 965–980.

ten Kate, A. 1972. Decomposition of linear programs by direct distribution. Econometrica 40: 883–898.

Kornai, J., and T. Liptak. 1965. Two-level planning. Econometrica 33: 141–169.

Kulikowski, R., L. Krus, K. Manczak, and A. Straszak. 1975. Optimization and control problems in large-scale systems. In Proceedings of IFAC 6th World Congress. Boston.

Lasdon, L. S. 1968. Duality and decomposition in mathematical programming. IEEE Transactions on Systems Science and Cybernetics 4: 86–100.

Lasdon, L. S. 1970. Optimization Theory for Large Systems. New York: Macmillan.

Mahmoud, M. S. 1977. Multilevel systems control and applications: A survey. IEEE Transactions on Systems, Man, and Cybernetics 7: 125–143.

Maier, S. F. 1974. A compact inverse scheme applied to a multicommodity network with resource constraints, pp. 179–203. In R. W. Cottle and J. Krarup (ed.), Optimization Methods for Resource Allocation. London: The English Universities Press.

Mandel', A. B. 1973. Internal prices in the control of industrial firms. (In Russian.) Ekonomika i matematicheskie metody 9: 500–513.

Murphy, F. H. 1973. Column dropping procedures for the generalized programming algorithm. Management Science 19: 1310–1321.

Pearson, J. D. 1971. Dynamic decomposition techniques, pp. 121–190. In D. A. Wismer (ed.), Optimization Methods for Large-Scale Systems. New York: McGraw-Hill.

Poliak, B. T., and N. V. Tret'iakov. 1972. On one iterative method of linear programming and its economic interpretation. (In Russian.) Ekonomika i matematicheskie metody 8: 740–751.

Schoeffler, J. D. 1971. Static multilevel systems, pp. 1–46. In D. A. Wismer (ed.), Optimization Methods for Large-Scale Systems. New York: McGraw-Hill.

Sekine, Y. 1963. Decentralized optimization of an interconnected system. IEEE Transactions on Circuit Theory 10: 161–168.

Simonnard, M. 1966. Linear Programming. Englewood Cliffs, New Jersey: Prentice-Hall.

Singh, M. G. 1977. Dynamical Hierarchical Control. Amsterdam: North-Holland.

Singh, M. G., S. Drew, and J. F. Coales. 1975. Comparisons of practical hierarchical control methods for interconnected dynamical systems. Automatica 11: 331–350.

Singh, M. G., M. F. Hassan, and A. Titli. 1976. Multilevel feedback control for interconnected dynamical systems using the prediction principle. IEEE Transactions on Systems, Man, and Cybernetics 6: 233–239.

Smith, N. J., and A. P. Sage. 1973. An introduction to hierarchical systems theory. Computers and Electrical Engineering 1: 55–71.

Tatjewski, P. 1975. Coordination by penalty function methods. In Proceedings of the Workshop Discussion on Multilevel Control. Warsaw.

Uzawa, H. 1958. Iterative methods for concave programming, pp. 154–165. In K. J. Arrow, L. Hurwicz, and H. Uzawa, Studies in Linear and Non-Linear Programming. Stanford, California: Stanford University Press.

Verina, L. F., and V. S. Tanaev. 1975. Decomposition approaches to the solution of mathematical programming problems. (In Russian.) Ekonomika i matematicheskie metody 11: 1160–1172.

Wolfe, P. 1967. Methods of nonlinear programming, pp. 97–131. In J. Abadie (ed.), Nonlinear Programming. Amsterdam: North-Holland.

4 Numerical Experiences with Dantzig–Wolfe Decomposition

4.1 ON THE UTILIZATION OF STRUCTURE IN SOLVING LINEAR PROGRAMMING PROBLEMS

Linear programming problems often have a special structure. This can be exploited in devising solution procedures that are particularly suited to those structures. One very well-known solution method applicable to a special problem structure is the transportation algorithm.

Multilevel solution methods are also often applied to exploit special structure. In particular, block-angular LP problem structures are likely candidates for an application of multilevel methods. However, other structures may also be considered for applications of multilevel methods, for instance LP problem structures where a subset of the constraints define the feasible region of a transportation problem (see subsection 3.3.4).

Among multilevel decomposition algorithms,* those that have been most widely applied are based on the Dantzig–Wolfe (DW) decomposition principle. For that reason, this chapter is devoted to a discussion of experiences gathered in applying such algorithms to various test problems.† Already at this point, it should be noted that those experiences have not been very positive. Partially as a consequence of that, so-called factorization methods have been developed as

* We distinguish here between multilevel and single-level decomposition algorithms, since the term "decomposition" is also sometimes used in the literature for procedures that (as we will argue) are essentially single-level.

† This should not be interpreted to mean that other decomposition algorithms have not been applied at all. For instance, in Chapters 6 and 8 of this volume two applications of the Benders decomposition algorithm are described. However, a number of studies of the DW decomposition method have been undertaken where emphasis was more on the performance of the method than on the solution to a particular real-world problem. Such studies are reviewed here. Several of the later chapters of this volume also discuss studies where the DW method was applied, but to real-world problems interesting in their own right (i.e., not to test problems).

an alternative to DW decomposition (see Müller-Merbach 1973, Orchard-Hays 1975, and Winkler 1974 for overviews of factorization methods). These methods may be regarded as extensions of the revised simplex method in that they attempt to store and manipulate the LP basis inverse in a compact form. This is also achieved by exploiting special structure. The best-known factorization method is generalized upper bounding (GUB) (Dantzig and Van Slyke 1967), which may be applied to problems with coefficient matrices of the type shown in Figure 4.1. This shows a very special block-angular matrix, where each subblock consists of a single row of ones (note that the extremal problem under DW decomposition has this structure).

GUB has been implemented with great success in a number of instances. One set of very similar extensions of GUB, "generalized GUB" (Lasdon 1970, pp. 340–356), "direct decomposition" (Müller-Merbach 1973), and the "block-product algorithm" (Orchard-Hays 1968, Chapter 12), allows the subblocks to be ordinary matrices (with more than one row and elements other than unity). In this case, one obtains a general block-angular structure. This means that generalized GUB and DW decomposition are alternative ways of exploiting the same block-angular structure.

The factorization methods are not regarded in this volume as multilevel methods. To do so would imply that the revised simplex method is also a multilevel method, since factorization methods may be regarded as extensions of the revised simplex method. For instance, GUB selects columns to enter and leave the basis in precisely the same sequence as the revised simplex method (or the ordinary simplex method, for that matter). Indeed, the factorization methods are referred to as "centralized" (implying single-level rather than multilevel) in Lasdon (1970, p. 304).

The conclusion is that there are several ways to exploit special structure in LP problems. The Dantzig–Wolfe method is one multilevel way. However, there are also single-level ways to exploit structure.

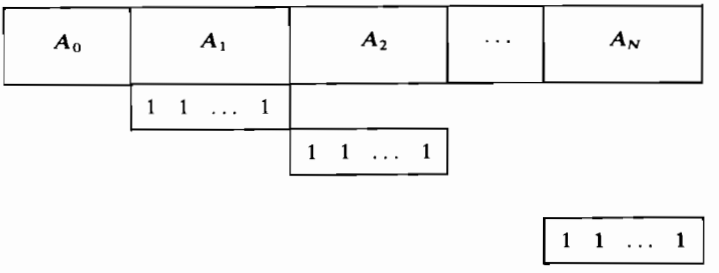


FIGURE 4.1 LP coefficient matrix with GUB structure.

4.2 TEST PROBLEM EXPERIENCES

As was mentioned above, DW is the decomposition method that has been applied most widely to LP problems. The following selected studies document numerical experiences with DW decomposition: Beale *et al.* (1965), Kutcher (1973), Ohse (1967), Schiefer (1973, 1976), Tcheng (1966) and Williams and Redwood (1974). In what follows, reference will be made to these studies. It is therefore of interest to mention a few facts about each of them.*

- Beale *et al.* (1965) consider a set of block-angular problems relating to oil field operations. No size data are given about the problems except that there were typically seven subblocks (infimal subproblems). In the course of this project, the DW method was programmed and incorporated as an option in an existing LP system, LP/90/94. Results concerning different tactics in implementing DW are given, as well as some comparisons with standard LP.
- Kutcher (1973) considers two block-angular problems: a small one with 60 rows and 72 activities, and a somewhat larger one with 187 rows and 382 activities. Results on the convergence of the algorithm, as a function of the number of decomposition iterations, are given, for a few different implementation tactics.
- Ohse (1967) compares three algorithms, DW, ordinary LP (using the revised simplex method with basis inverse in explicit form), and direct decomposition (a factorization method). Twelve different block-angular test problems are considered, typically of size 120×150, with five to ten subblocks. Computer programs were written for all algorithms in ALGOL. Apparently, the problems considered were so small that they could be solved in core (i.e., without the transfer of data to and from peripheral memory units), with all three methods.
- Schiefer (1973) and Schiefer (1976) refer to the same investigation. One example problem with block-angular structure is considered. The size is $907 \times 1{,}265$. There are 32 subblocks, each of size 28×39. (However, in most of the runs four infimal subproblems were used; see also below.) This problem was solved using some different implementation tactics, and a comparison with ordinary LP was made as well. For this investigation, a DW routine was built around an already existing LP code.
- Tcheng (1966) considers a single problem that arose in conjunction with forest management. The size is around $1{,}200 \times 28{,}000$. The structure of the problem is peculiar in that almost all the constraints are of the type $\sum x_{ij} = 1$, where the summations range over disjoint subsets of the variables. That is, the constraint matrix looks like that in Figure 4.2. Let the lower block in Figure 4.2 be denoted B. In passing, it may be observed that this problem could have been

* Additional test studies of DW decomposition have undoubtedly been performed, but the results may not have been published. For instance, Malkov (1969) mentions that some experimental DW codes have been written in the USSR, but he gives no test results.

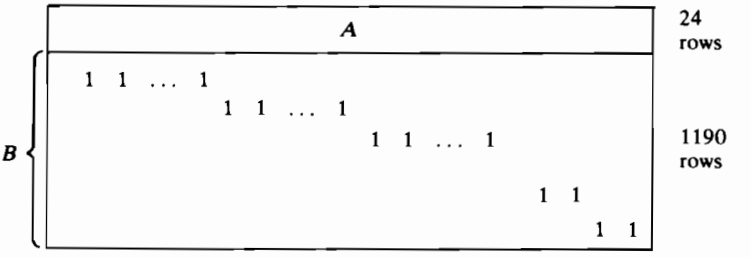


FIGURE 4.2 Tcheng's problem structure.

solved by GUB, since block B has the structure required for that. However, GUB was not available (or at least not widely known) at the time. Dantzig–Wolfe decomposition was applied, with only one infimal subproblem, constructed from block B. Block A was used for the restricted master problem. A DW program was written in FORTRAN. The problem was solved on an IBM 7044 computer with 32K core storage. Double precision was used.

- Williams and Redwood (1974) solve two problems arising in the food industry. The sizes are 358×804 and $1{,}805 \times 3{,}236$, and the number of subblocks 13 and 4, respectively. There were 24 and 132 coupling constraints, respectively. These problems were solved by DW and ordinary LP. The IBM MPSX system was utilized. For DW, this involved using PL/1 procedures to build a decomposition routine around the MPSX system.

As mentioned earlier, the experiences with DW have not been altogether positive. More specifically, DW is often both time-consuming and cumbersome. The time needed to run a given problem on the computer, assuming that a DW program is on hand, is often inordinately great. Several authors have reported that DW converges only slowly towards an optimal solution to the extremal problem; i.e., the objective function value of the restricted master problem converges slowly in the adjustment phase. Or, more precisely, there is often rapid progress in the earlier iterations, but later progress is quite slow. One probably fairly typical sample problem is depicted in Figure 4.3, which is adapted from a table in Kutcher (1973, p. 514). This problem (the larger of the two considered by Kutcher) required 18 iterations to reach an optimum. The objective function value of the restricted master problem increases rapidly in iterations 4–8, but after only slowly. The behavior of the upper bound on the objective function value is also depicted in the graph.

An even more extreme case of slow convergence is given by Tcheng (1966). Table 4.1 gives the objective function value for different number of iterations. After 960 iterations, around 800 minutes of computation time had been used up. The computations were then stopped without an apparent optimal solution. It may be noted that between 930 and 960 iterations, there was actually a decrease in objective function value. This is contrary to the theoretical

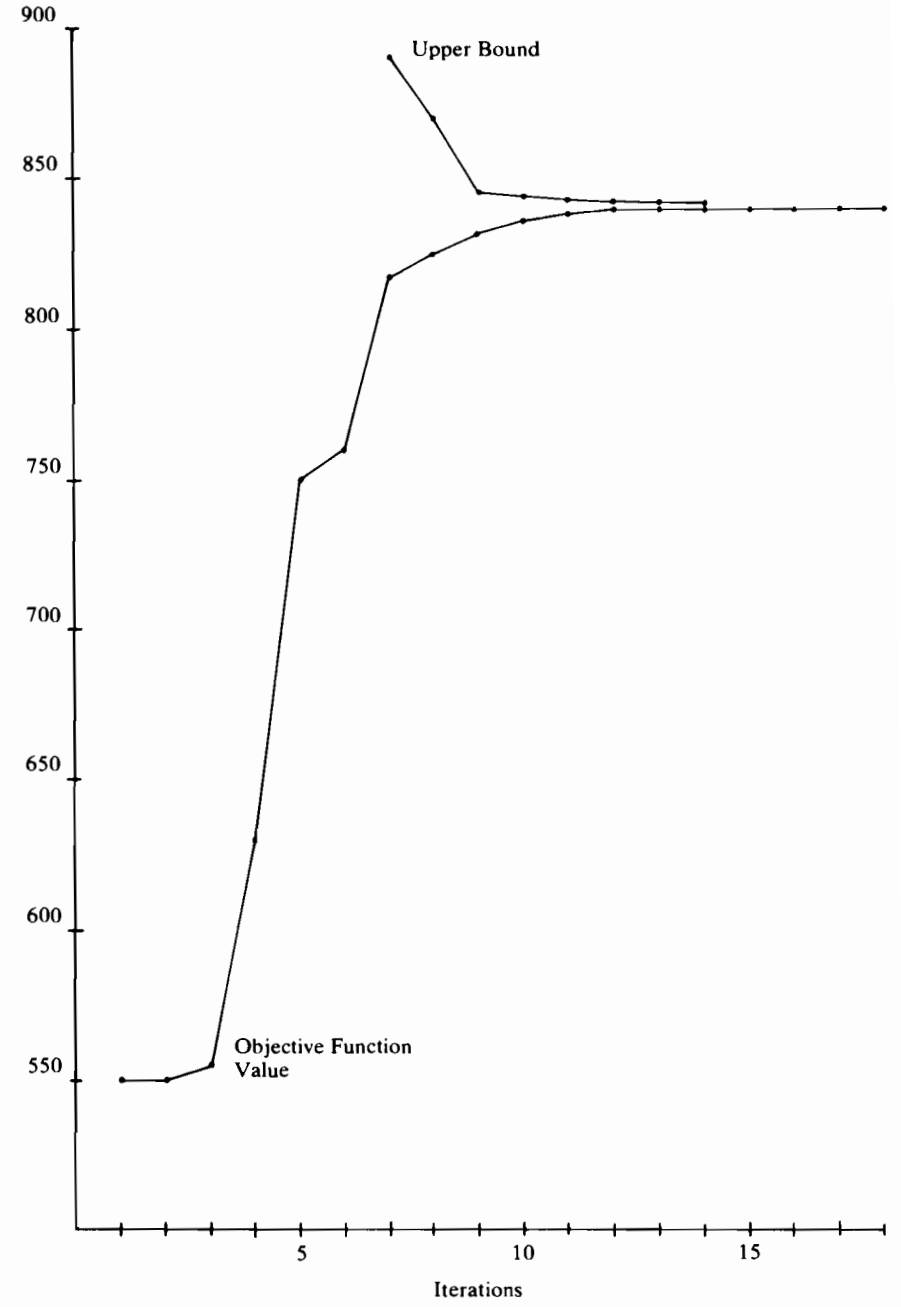


FIGURE 4.3 Convergence of Dantzig–Wolfe method in sample problem. (After iteration 14, the difference between upper bound and objective function is less than 0.5.) Source of data: Kutcher (1973, p. 514).

TABLE 4.1 Convergence of Dantzig–Wolfe Method in Tcheng's Problem

Number of Iterations	Objective Function Value	Increase per 30 Iterations
30	1,452,346	
60	1,489,466	37,120
90	1,498,481	9,015
120	1,502,308	3,827
150	1,503,698	1,390
⋮		
810	1,507,289	
840	1,507,301	12
870	1,507,307	6
900	1,507,321	14
930	1,507,329	8
960	1,506,028	−1,301

SOURCE: adapted from Tcheng (1966, p. 63).

properties of the DW method, and is probably the result of rounding-off errors (Tcheng 1966, pp. 101–103).

Since convergence is often very slow in later decomposition iterations, it becomes desirable to stop before reaching optimality. A common procedure is to stop when the difference between the current restricted master-problem objective-function value and the best bound obtained so far is smaller than some specified constant. This procedure was used by Schiefer (1973, 1976), among others.

Apart from being time-consuming in many cases, DW is also somewhat cumbersome to implement. One reason for this is that standard DW codes are usually not available. One exception is the LP/90/94 system, which did include DW decomposition facilities as a result of the system development work undertaken by Beale and his colleagues and reported in Beale *et al.* (1965). However, the LP/90/94 system is fairly old by now, and it appears that it has been withdrawn from the market. In general, the user of DW decomposition must build his own system, using some existing LP code as a central component, as was done by Williams and Redwood (1974) and by Schiefer (1973, 1976). For instance, in Schiefer (1973, 1976), the system was constructed to operate in the manner shown in Figure 4.4. For the optimization of the restricted master problem and the infimal subproblems, an existing LP code could be utilized. The remainder of the system had to be constructed by the author.

Another reason why DW is cumbersome to implement is that the solution finally obtained relates to the restricted master problem, not to the original problem. The solution to the original problem must then be recovered in the

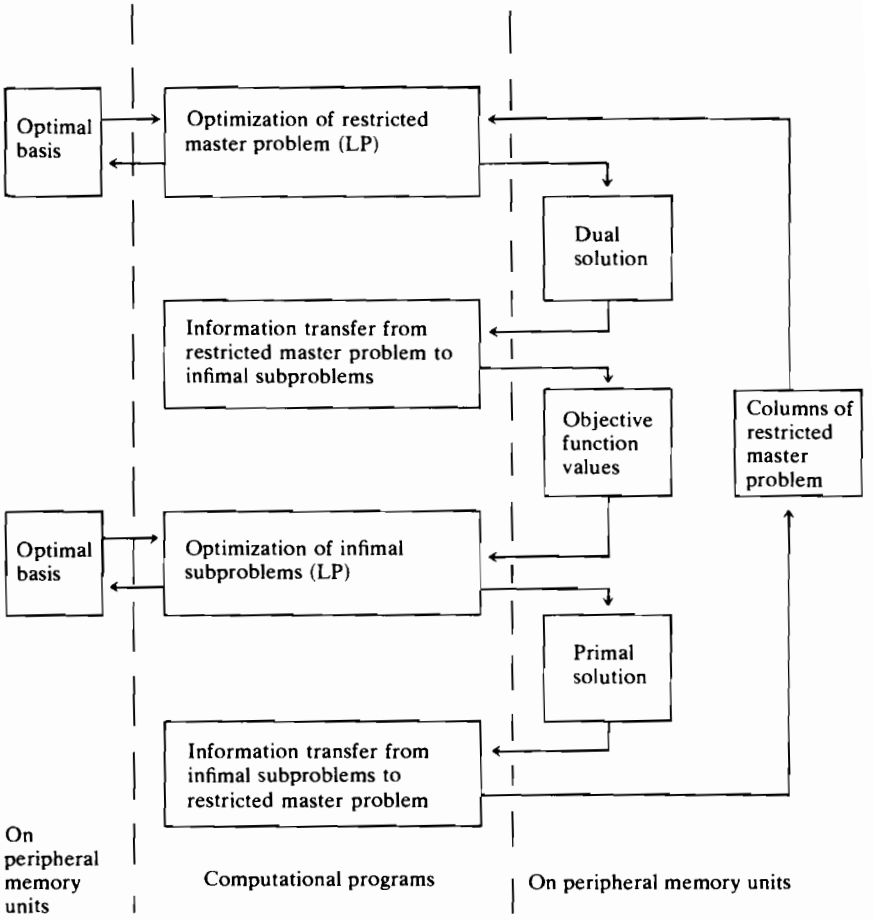


FIGURE 4.4 Buildup of system for Dantzig–Wolfe decomposition. After Schiefer (1973, p. 11).

execution phase. Suppose the original problem is the following block-angular one:

$$\begin{aligned} \text{Maximize} \quad & c_1x_1 + \cdots + c_nx_n \\ \text{s.t.:} \quad & A_1x_1 + \cdots + A_nx_n \le a, \\ & B_1x_1 \qquad\qquad \le b_1, \\ & \qquad\qquad \ddots \\ & \qquad\qquad B_nx_n \le b_n, \\ & x_1 \cdots x_n \ge 0. \end{aligned} \tag{4.1}$$

Suppose for simplicity that each set $\{x_j | B_jx_j \le b_j,\ x_j \ge 0\}$ is bounded and that the

extreme points are denoted x_j^p. The equivalent extremal problem is then (see subsection 3.3.6)

$$\begin{aligned} &\text{Maximize} && \textstyle\sum_p (c_1 x_1^p)\lambda_1^p + \cdots + \sum_p (c_n x_n^p)\lambda_n^p \\ &\text{s.t.:} && \textstyle\sum_p (A_1 x_1^p)\lambda_1^p + \cdots + \sum_p (A_n x_n^p)\lambda_n^p \leq a, \\ & && \textstyle\sum_p \lambda_j^p = 1 (j = 1 \ldots n), \lambda_j^p \geq 0. \end{aligned} \tag{4.2}$$

By DW decomposition, one obtains some solution (optimal or not) to (4.2). However, that is not what one is usually interested in. Rather, one wants a solution to (4.1). In textbook treatments of DW decomposition, it is often suggested that a solution to (4.1) be recovered as

$$x_j = \textstyle\sum_p \bar{\lambda}_j^p x_j^p,$$

where the solution to (4.2) is denoted $\bar{\lambda}_j^p$. However, this method can usually not be used in practice (Beale 1968, p. 168; Beale *et al.* 1965, p. 14; Orchard-Hays 1968, p. 245; 1973, p. 162), as was pointed out above, in section 3.3.6. Using it would require that the definitions of all x_j^p be stored, "an intolerably enormous data processing task" (Orchard-Hays 1973, p. 162). Instead Beale *et al.* obtain the final solution to (4.1) in the execution phase by considering an infimal subproblem of the following kind for each index $j = 1 \ldots n$:

$$\begin{aligned} &\text{Maximize} && c_j x_j \\ &\text{s.t.:} && A_j x_j \leq \textstyle\sum_p (A_j x_j^p)\bar{\lambda}_j^p, \\ & && B_j x_j \leq b_j, \\ & && x_j \geq 0. \end{aligned}$$

A similar procedure is suggested by Orchard-Hays (1973). But this means that a separate program block must be added, in the execution phase, for the recovery of the final solution to the original problem.

Some more specific results from the investigations listed earlier will now be considered. There are two kinds of results. In the first place, it is of interest to compare DW with other methods, in particular ordinary LP. Results of a few such comparisons will be given. In the second place, the tactics for implementing DW can be varied from case to case. In so doing, one is, in effect, comparing different versions of DW. Results relating to different tactics in implementing DW will also be given.

As for comparisons of DW with ordinary LP, Beale *et al.* (1965, p. 18) report that some savings in running time were realized by using DW for problems with 300–500 rows. For instance, one 450-row problem was solved in 40 minutes by the simplex method but in 37 using DW. A larger problem was solved in 5 hours using ordinary LP, while solution of a similar problem by DW required only 2 hours. This comparison is unduly flattering to DW, though, since the DW solution

was obtained after extensive experimentation with implementation tactics, whereas the LP solution was obtained from scratch (i.e., without the specification of a reasonable starting basis).

Ohse (1967) found that DW performed about as well as ordinary LP. In his test problems, DW was superior to ordinary LP in some cases but inferior in others. However, DW was inferior to direct decomposition. This comparison is not entirely fair, as direct decomposition is not usually a natural alternative to DW. Standard codes for direct decomposition apparently do not exist, and it is a rather complex method to implement.

Schiefer (1973, 1976) divided the total running time in his DW experiments into two parts: "calculation time" and "storage time" (a specification of CPU time was not possible; see Schiefer 1976, p. B9). In calculation time, the DW runs (9 runs) performed about as well as ordinary LP (two runs, with different starting solutions). Calculation time was typically around 4–5 minutes. However, in addition there were heavy storage times for the DW runs, typically around 20–30 minutes. In other words, the transfer of various problem parts to and from peripheral memory units turned out to be very time-consuming. For the ordinary LP runs, storage time was zero (which presumably means that the test problem could be solved by ordinary LP in core).

Williams and Redwood (1974) performed three different runs of their smaller model with ordinary LP and three with DW. They conclude that DW just about breaks even with ordinary LP for problems that size, if the LP procedure can be started off with a reasonable initial solution. One run of their larger model was performed with both solution methods. In this case, DW took 400 CPU seconds, and ordinary LP, 800. However, this may again be somewhat too flattering to DW, since the LP solution was obtained from an all-slack starting basis, whereas the DW procedure was begun with four different solutions to each infimal subproblem, to form columns for the restricted master problem. Additionally, the ultimate solution to the original problem (as opposed to the extremal problem) was never recovered. For the problem situation at hand, the extremal problem did yield meaningful information, and a recovery of the original problem solution was not necessary. That is, the execution phase was deleted. The DW computations were terminated when the improvement in restricted master-problem objective-function value in successive iterations was less than 0.5 percent. This occurred after three iterations in all four cases. This means that optimal solutions were not obtained using DW. The difference between the solution values obtained using DW and the true optimal solution values, obtained through ordinary LP, were quite small. The authors conclude that DW is worthwhile for the larger problem.

One may conclude at this point that, with regard to computer time usage, there is no convincing evidence that DW performs substantially better than ordinary LP. In addition to computer time usage for running the particular problems one is interested in, one must also consider the effort required for

building a DW decomposition system. As mentioned earlier, standard programs for DW are usually not available, so the user must create his own. The upshot is that if a standard LP code is available that can handle the problem under consideration, it is not worthwhile to attempt to use DW. This recommendation has, in fact, been stated by several authors (e.g., Kornai 1973, p. 526).

However, there may nevertheless exist problem situations that exceed the limits of existing LP codes. Even in 1973 there were a few standard programs available that were capable of handling problems with more than 10,000 rows. Nonetheless, one can imagine that even larger problems may need to be solved, and in that case some method other than ordinary LP must be used. For instance, the test problem considered by Schiefer (1973, 1976) could be solved directly, by the simplex method. However, it was foreseen that considerably expanded versions of that problem would later be constructed, and standard LP programs would then no longer suffice. In this type of situation, DW may well be a reasonable choice, since it is a conceptually simple procedure. Dantzig–Wolfe systems can be built around existing LP codes, whereas factorization methods, for instance, are messy and often more difficult to program*. For that reason, there is some interest in examining results on different implementation tactics for DW, as a guide for those situations where DW must be used. This brings us to the second kind of results mentioned earlier.

The size of the problem to be solved influences the time required for its solution. This is obviously true in general for any solution method, but for DW decomposition the number of coupling constraints is mentioned as particularly critical (Beale 1968, p. 171; Dantzig and Van Slyke 1971, p. 95). Hence, it is a good idea in the modeling stage to try to keep down the number of coupling constraints.

Assume, however, that a particular LP problem has been formulated and is to be solved by DW decomposition. There are then certain options to consider, among them the following:

1. How many columns from each infimal subproblem should be used to construct the initial restricted master problem?
2. How many, and which, proposals from each infimal subproblem should be added to the restricted master problem at each iteration?
3. Should nonbasic columns be deleted from the restricted master problem?

* It may be mentioned in this connection that a later study by Ohse (1971) contains test problem runs with DW and three different factorization methods: generalized GUB, Ohse's own dual algorithm, and Rosen's algorithm (this latter algorithm is classified as a factorization method in Winkler 1974). Twenty-five test problems were solved, with numbers of rows ranging from 110 to 315. The algorithms were programmed by the author. In-core storage of the whole problem was not used. Rather, peripheral memory units were used as well, and the total computation time includes access to these peripheral units. In total computation time (summed over all 25 test problems), DW was superior to generalized GUB and to Rosen's algorithm, but somewhat inferior to Ohse's algorithm (Ohse 1971, pp. 58–78). A comparison with ordinary LP was not made.

4. How many infimal subproblems should one utilize? Should some infimal subproblem(s) be joined with the restricted master problem?

These points will be considered in turn.

1. It is obviously a good idea to construct the first restricted master problem utilizing several, rather than just one (or even zero), columns pertaining to each infimal subproblem, if several such columns are available or can be generated easily (Orchard-Hays 1968, p. 245), and this has been done in several studies. Williams and Redwood (1974) initialized their DW runs with four columns from each infimal subproblem. Beale *et al.* (1965, pp. 15–16) wrote a special generator program to provide good sets of columns from which to construct the first restricted master problem. Schiefer (1976, pp. B11–B12) and Kutcher (1973, pp. 517–518) demonstrate that much more rapid progress can be obtained in the early decomposition iterations if the restricted master problem is started off with several columns from each infimal subproblem, compared to the situation where only one (or zero) column from each infimal subproblem is available at the outset.

Related to the question of the number of initial columns is the "goodness" of those columns. A good initial solution to the restricted master problem is one for which the objective function value is close to the true optimal objective function value, and for which the values of the dual multipliers pertaining to the coupling constraints are not too different from the optimal multiplier values. The importance of a good initial solution to the restricted master problem is pointed out by Beale *et al.* (1965, p. 15). On the other hand, though, Schiefer (1976, pp. B11–B12) experimented with some different initial restricted master problem solutions for his test problem, characterized by different degrees of goodness. He found that the degree of goodness of the initial solution was not a precise predictor of the rate of convergence, in other words, it did not seem to matter much.

2. The usual convention under the DW method is that in each iteration, no more than one column from each infimal subproblem is added to the restricted master problem. That column is derived from the optimal solution to the corresponding subproblem in that iteration. However, Beale *et al.* (1965) found that it was more efficient to submit several columns from each infimal subproblem to the restricted master problem in each iteration. All of these columns except one would then correspond to basic, nonoptimal solutions to the infimal subproblem traversed on the way to the optimal solution. Additionally, they found that it was sensible not to obtain optimal solutions to the infimal subproblems at all. That is, the infimal subproblems were not solved to optimality in the early iterations of the algorithm. Rather, they were cut off before that, meaning that for each infimal subproblem, a set of columns pertaining to basic, nonoptimal infimal subproblem solutions was added to the

restricted master problem. Without these features, they found that convergence was so slow that DW was "more or less useless" (Beale *et al.* 1965, p. 15).

3. According to Dantzig's own description, nonbasic columns in the restricted master problem may be dropped (Dantzig 1963, p. 453; see Orchard-Hays 1968, p. 252). But if this is done, they may have to be generated anew at some later iteration. On the other hand, if all columns are kept, the restricted master problem may eventually become quite sizable in terms of number of variables. Schiefer's investigation sheds some light on the trade-off involved here. He reports that restricted master problem columns typically had a useful life of no more than 15 iterations or so after becoming nonbasic, meaning that if a column is nonbasic 15 iterations after it was pivoted out of the basis, it can usually be dropped with no risk of its needing to be re-created at a later iteration (Schiefer 1973, p. 12; Schiefer 1976, p. B7).

4. Suppose some block-angular LP problem has a constraint matrix as displayed in Figure 4.5. In this situation, it is intuitively most natural to use the coupling constraint block $[A_1, A_2, A_3, A_4]$ for the restricted master problem, and use four infimal subproblems, one for each subblock B_1–B_4. However, other choices are also possible. One could, for instance, form a restricted master problem out of the blocks $[A_1, A_2, A_3, A_4]$ and B_1 and then use three infimal subproblems, corresponding to the subblocks B_2–B_4*. Beale states that such an arrangement (i.e., incorporating one or several subblocks in the restricted master problem) is often a good idea (Beale *et al.* 1965, p. 14; Beale 1968, p. 169), as it may provide for more realistic and stable multiplier values for the common rows. The restricted master problem will require more time in each iteration, but fewer iterations may be needed in total. Schiefer (1973, 1976) performed two test runs with one and two subblocks, respectively, incorporated in the restricted master problem. The results were better than in the situation where no subblocks were put into the restricted master problem (Schiefer 1976, p. B13). Convergence was much more rapid in early iterations.

* If one forms the restricted master problem out of the blocks $[A_1, A_2, A_3, A_4]$ and B_1 and uses three infimal subproblems, the resulting extremal problem may be written as follows, using the notation of section 3.3.6; see also problem (3.17) of that section:

$$\text{Maximize} \quad c_1x_1 + \sum_{j=2}^{4}\left(\sum_{p=1}^{P(j)} w_j^p\lambda_j^p + \sum_{r=1}^{R(j)} \tilde{w}_j^r\delta_j^r\right)$$

$$\text{s.t.:} \quad A_1x_1 + \sum_{j=2}^{4}\left(\sum_{p=1}^{P(j)} L_j^p\lambda_j^p + \sum_{r=1}^{R(j)} \tilde{L}_j^r\delta_j^r\right) \le a,$$

$$B_1x_1 \le b_1,$$

$$\sum_{p=1}^{P(j)} \lambda_j^p = 1 (j = 2, 3, 4),$$

$$x_1 \ge 0, \lambda_j^p \ge 0, \delta_j^r \ge 0.$$

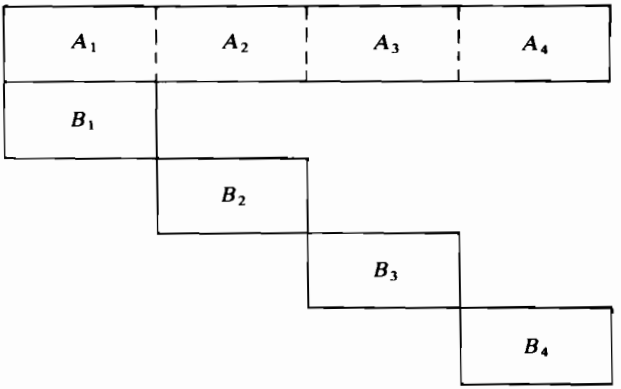


FIGURE 4.5 Block-angular LP structure.

Heuristically, it is easy to see why incorporating one or several subblocks in the restricted master problem may have a good effect on convergence. The restricted master problem may be thought of as iteratively collecting information from the infimal subproblems, as computations proceed in the adjustment phase. If a subblock is put into the restricted master problem, then there is that much more information on hand at the outset.

Once a division of constraint blocks between restricted master problem and infimal subproblems has been decided upon, the question still remains of how many infimal subproblems to utilize. In the sample problem sketched above, suppose one has decided to form the restricted master problem out of the blocks $[A_1, A_2, A_3, A_4]$. One may then, for example, form four infimal subproblems, one for each subblock B_1–B_4; or two infimal subproblems, one for B_1 and B_2, and one for B_3 and B_4; or one infimal subproblem, encompassing all of the subblocks B_1–B_4. The number of infimal subproblems obviously affects the number of convexity rows in the restricted master problem, and also the number of new columns submitted to the restricted master problem in each iteration. Madsen (1973) has attempted to derive analytical rules for the decision on the optimal number of infimal subproblems. Under a fairly wide range of circumstances, he found that a maximum number (i.e., as large as possible; four in the above case) is optimal in terms of total computation time. This conclusion is supported by Schiefer's investigation; Schiefer (1976) solved his example problem with a varying number of infimal subproblems (one, four and eight). With decreasing number of infimal subproblems, there was an increase in total computation time (Schiefer 1976, p. B10). Kutcher (1973) solved the smaller one of his two test problems in two versions: with 32 and 8 infimal subproblems. In the former case, an optimum was reached after six iterations, and in the latter case after eight (Kutcher 1973, p. 507).

Finally, it should be noted that Beale *et al.* (1965) found it useful to vary the implementation tactics over the course of the adjustment phase. That is, during early iterations, many columns from each infimal subproblem were submitted to the restricted master problem. During intermediate iterations, that number was diminished, and then increased again during later iterations. Similarly, during early iterations the infimal subproblems were cut off before optimality was reached, but this was not done during later iterations. These choices were made as the adjustment phase progressed. That is, one was able to follow the progression of the computations on an on-line printer and make decisions on tactics as the calculations proceeded. This sort of interaction with the algorithm is reminiscent of "man–machine planning" (Kornai 1969), which is discussed in Chapter 5.

REFERENCES

Beale, E. M. L. 1968. Mathematical Programming in Practice. London: Pitman.

Beale, E. M. L., P. A. B. Hughes, and R. E. Small. 1965. Experiences in using a decomposition program. Computer Journal 8:13–18.

Dantzig, G. B. 1963. Linear Programming and Extensions. Princeton, New Jersey: Princeton University Press.

Dantzig, G. B., and R. M. Van Slyke. 1967. Generalized upper bounding techniques. Journal of Computer and System Sciences 1:213–226.

Dantzig, G. B., and R. M. Van Slyke. 1971. Generalized linear programming, pp. 75–120. In D. A. Wismer (ed.), Optimization Methods for Large-Scale Systems. New York: McGraw-Hill.

Kornai, J. 1969. Man–machine planning. Economics of Planning 9:209–234.

Kornai, J. 1973. Thoughts on multi-level planning systems, pp. 521–551. In L. M. Goreux and A. S. Manne (ed.), Multi-Level Planning: Case Studies in Mexico. Amsterdam: North-Holland.

Kutcher, G. P. 1973. On the decomposing price-endogenous models, pp. 499–519. In L. M. Goreux and A. S. Manne (ed.), Multi-Level Planning: Case Studies in Mexico. Amsterdam: North-Holland.

Lasdon, L. S. 1970. Optimization Theory for Large Systems. New York: Macmillan.

Madsen, O. B. G. 1973. The connection between decomposition algorithms and optimal degree of decomposition, pp. 241–250. In D. M. Himmelblau (ed.), Decomposition of Large-Scale Problems. Amsterdam: North-Holland.

Malkov, U. Kh. 1969. A survey of programs for solving the general linear programming problem. (In Russian.) Ekonomika i matematicheskie metody 5:594–597.

Müller-Merbach, H. 1973. Upper-bounding technique, generalized upper-bounding technique, and direct decomposition in linear programming: A survey on their general principles including a report about numerical experience, pp. 167–180. In D. M. Himmelblau (ed.), Decomposition of Large-Scale Problems. Amsterdam: North-Holland.

Ohse, D. 1967. Numerische Erfahrungen mit zwei Dekompositionsverfahren der linearen Planungsrechnung. (Numerical experiences with two linear programming decomposition procedures, in German.) Ablauf- und Planungsforschung 8 (2):289–301.

Ohse, D. 1971. Ein dualer Dekompositionsalgorithmus zur Lösung Blockangularer Probleme der linearen Planungsrechnung. (A Dual Decomposition Algorithm for Solving Block-Angular LP Problems, in German.) Ph.D. dissertation. Technische Hochschule Darmstadt.

Orchard-Hays, W. 1968. Advanced Linear-Programming Computing Techniques. New York: McGraw-Hill.

Orchard-Hays, W. 1973. Practical problems in LP decomposition and a standardized phase I decomposition as a tool for solving large-scale problems, pp. 153–166. In D. M. Himmelblau (ed.), Decomposition of Large-Scale Problems. Amsterdam: North-Holland.

Orchard-Hays, W. 1975. Factoring LP block-angular bases, pp. 75–92. Mathematical Programming Study 4: Computational Practice in Mathematical Programming. Amsterdam: North-Holland.

Schiefer, G. 1973. Zur Anwendung linearer Dekomposition. (On the use of linear decomposition, in German.) Paper presented at DGOR Annual Conference, Karlsruhe.

Schiefer, G. 1976. Lösung grosser linearer Regional-planungsprobleme mit der Methode von Dantzig und Wolfe. (Solution of large regional-planning problems with the method of Dantzig and Wolfe, in German.) Zeitschrift für Operations Research 20:B1–B16.

Tcheng, T. H. 1966. Scheduling of a Large Forestry-Cutting Problem by Linear Programming Decomposition. Ph.D. dissertation. University of Iowa.

Williams, H. P., and A. C. Redwood. 1974. A structured programming model in the food industry. Operational Research Quarterly 25: 517–527.

Winkler, C. 1974. Basis Factorization for Block-Angular Linear Programs: Unified Theory of Partitioning and Decomposition Using the Simplex Method. RR-74-22. Laxenburg, Austria: International Institute for Applied Systems Analysis.

5 National and Regional Economic Planning

5.1 INTRODUCTION AND OVERVIEW

In this chapter, we are concerned with multilevel models and methods for national and regional economic planning. National economic planning refers here to the planning of production, foreign trade, and investments for an entire economy. As will be seen, the total planning problem may be represented as a large mathematical programming model or a system of such models. The variables refer to production, investments, and so on. The restrictions derive from physical capacity limits (e.g., manpower limits), and also from the input–output relations of the economy. The objective function could, for instance, be a maximization of revenue from foreign trade.

There are at least two reasons why multilevel methods are of importance for national economic planning. First, the number of variables in a national economic planning problem is overwhelming. Pugachev (1974, p. 477), for example, mentions that for the Soviet Union, the nomenclature (list of commodities) consists of around 2×10^6 items. Assuming that each commodity can be manufactured in 50 different regions and transported to 50 different regions and that it can be produced in each location by 10 different methods, the total number of variables for a 10-year plan will be $2 \times 10^6 \times 50^2 \times 10 \times 10$. It is clearly impossible to solve, or even formulate, problems of this size in a single-level fashion. Some other approach, such as multilevel methods or aggregation (or a combination of both), must be utilized.

Second, in the Eastern European planning literature, the idea is often advanced that the whole national economy is to be regarded as a hierarchical system, where different hierarchical levels may be represented by the central planning level, industrial sectors, and individual firms. This idea is frequently encountered in the Soviet literature (e.g., Baranov *et al.* 1971; Fedorenko 1974; Katsenelinboigen and Faerman 1967; Kantorovich 1976), and also in

the writings of Kornai (e.g., Kornai 1975; Kornai and Liptak 1965). Since the economy is viewed as a multilevel system, national economic planning could also be carried out as an institutional multilevel process, where the total planning task is divided between organizations on different levels and where messages are physically exchanged between them in the adjustment phase (see section 2.2.2). Some Soviet economists apparently envision the performance of national economic planning in this manner; this has also been noted by Western observers of Soviet mathematical economics (Ellman 1973, pp. 128–133; Zauberman 1975, pp. 35–36). The Dantzig–Wolfe decomposition principle is sometimes taken as a model of how the economy could eventually function (e.g., by Kantorovich 1976, p. 209).

The two case studies of multilevel national economic planning described in this chapter, one dealing with Hungary and the other with Mexico, represent somewhat different approaches, and a comparison between the two is instructive. It should be mentioned here that multilevel methods were used in these two cases as purely computational tools. That is, an institutional multilevel planning process was not attempted.

Additional references on national economic planning by multilevel methods could easily be cited. For instance, Kronsjö (1963) and Trzeciakowski (1973) formulate LP models for foreign trade optimization and then indicate how Dantzig–Wolfe decomposition can be applied. A rather large experimental research effort is the work in the USSR by Pugachev and his associates on "multistage optimization" (Martynov and Pitelin 1969; Pugachev *et al.* 1972, 1973; Fedorenko 1975, Chapters 1–4). Hence the two case studies described here do not exhaust the literature on multilevel national economic planning. It is difficult, however, to find well-documented, actually implemented cases, and in this respect the Hungarian and Mexican studies are unusual.

This chapter also considers regional economic planning. Obviously, the difference between regional and national planning is not a sharp one. A Soviet study in multilevel regional planning is presented in section 5.4. Because of the size of the Soviet Union, that case study could well have passed for "national" in a smaller country.

It may be pointed out here that the case studies of this chapter (the Hungarian and Mexican case studies in national economic planning and the Soviet study in regional planning) hold a common methodological interest: they illustrate heuristic multilevel approaches.

5.2 MULTILEVEL NATIONAL ECONOMIC PLANNING IN HUNGARY

5.2.1 THE APPLICATION OF THE KORNAI–LIPTAK METHOD TO A NATIONAL ECONOMIC PLANNING PROBLEM

Probably the most extensive attempt to apply multilevel methods to actual, real-world national economic planning problems has been undertaken in Hungary, by Kornai and his associates. This section draws upon Kornai's work (Kornai 1965, 1969a, 1969b, 1975; Kornai and Liptak 1965). Ganczer (1973) also offers a discussion of the use of multilevel planning methods in Hungary.

An early step in the use of formalized methods for national economic planning in Hungary was the construction of sectoral LP models (Kornai 1975, Chapters 5 and 6). A sectoral LP model could, for instance, schedule production and investment activities in a certain sector of the economy, such as the cotton fabrics industry. The objective function could be a minimization of total societal cost or a maximization of foreign exchange revenue, subject to meeting production goals stated as plan directives.

Rather naturally, the idea then emerged that one could construct larger models encompassing several sectors. In such a model, there would be certain sectoral constraints, restricting activity levels in only one sector, but also other constraints, restricting activity levels in all sectors taken together. These latter constraints could, for example, refer to total manpower availabilities in the entire economy. One would, in effect, obtain a block-angular LP problem. One could imagine as associated with such a problem organizational units on two different hierarchical levels. On the first (higher) level, there would be the "center" (in actual practice the National Planning Bureau). On the second level, there would be industry sectors, corresponding to ministries or sections of ministries. Each sector is responsible for the production, investment, and foreign trade relating to a specific product group. The coupling constraints may then be thought of as pertaining to the center, and the remaining (subblock) constraints to different sectors.

In this section, we will indicate in greater detail how such a block-angular LP problem can be formulated (Kornai and Liptak 1965). Let i and j index sectors $(i, j = 1 \ldots n)$. A T-year horizon is considered. $t = 1 \ldots T$ indexes years. The center is constrained by the following two sets of economic policy figures:

$Q_{it} (i = 1 \ldots n;\ t = 1 \ldots T)$: the final consumption of product group i by individuals and public bodies foreseen for year t

W_t $(t = 1 \ldots T)$: the manpower availability in year t

The center issues three types of directives to the sectors:

r_{it} $(i = 1 \ldots n;\ t = 1 \ldots T)$: the amount of product group i that sector i is required to provide in year t (the means of providing the amount r_{it} include production and imports, as will be seen later)

z_{ijt} $(i = 1 \ldots n;\ j = 1 \ldots n;\ j \neq i;\ t = 1 \ldots T)$: the amount of product group j assigned to sector i in year t; to be used as input in production and other activities in sector i

w_{it} $(i = 1 \ldots n;\ t = 1 \ldots T)$: the amount of manpower assigned to sector i in year t

The center faces the following rather obvious constraints in making decisions on r_{it}, z_{ijt}, and w_{it}:

$$r_{it} - \sum_{\substack{j=1 \\ j \neq i}}^{n} z_{jit} = Q_{it} \qquad (i = 1 \ldots n;\ t = 1 \ldots T), \tag{5.1}$$

$$\sum_{i=1}^{n} w_{it} = W_t \qquad (t = 1 \ldots T), \tag{5.2}$$

$$r_{it} \geq 0,\ z_{ijt} \geq 0,\ w_{it} \geq 0. \tag{5.3}$$

Additionally, the following set of constraints is imposed:

$$r_{it} \leq R_{it} \qquad (i = 1 \ldots n;\ t = 1 \ldots T). \tag{5.4}$$

Constraint (5.4) merely serves to bound the set of feasible r_{it} and z_{ijt} choices. It has no real economic meaning. It will be seen presently that (5.1), (5.2), and (5.4) constitute the coupling constraints of a block-angular LP problem.

Turning now to the sectors, sectoral activity levels are denoted by x_{ikt}, where, as before, i indexes sectors and t indexes years. k belongs to certain index sets: $k \in \mathcal{R}_i$ or $k \in \mathcal{E}_i$ or $k \in \mathcal{M}_i$. $\mathcal{R}_i$ is the set of production activities in sector i. One member of that set could, for instance, be production in a particular factory belonging to the sector. $\mathcal{E}_i$ is the set of export activities for sector i (different export markets). $\mathcal{M}_i$ is the set of import activities (import markets).

Additionally, each sector i has certain investment activities available. The levels are expressed by the variables x_{ik}, $k \in \mathcal{I}_i$, where $\mathcal{I}_i$ is the set of possible such activities for sector i. The investment variables are not indexed by t, since the same physical project started in two different years is regarded as two different investment activities.

Finally, each sector i also has at its disposal an unbounded, very high-cost artificial import activity, the levels of which are given by x_{iot} $(t = 1 \ldots T)$. The x_{iot} should be regarded as artificial variables; they ensure that the sectoral subproblems are always feasible, irrespective of the choices of r_{it}, z_{ijt}, and w_t.

The first set of constraints on each sector i says that the sector must in year t provide at least that amount r_{it} of its product group that is specified by the center:

$$\sum_{k \in \mathscr{R}_i} x_{ikt} - \sum_{k \in \mathscr{E}_i} x_{ikt} + \sum_{k \in \mathscr{M}_i} x_{ikt} + \sum_{k \in \mathscr{I}_i} f_{ikt} x_{ik} + x_{iot} \geq r_{it} \qquad (t = 1 \ldots T). \tag{5.5}$$

Expressed verbally, (5.5) says that amounts produced (in the various production activities) minus amounts exported plus amounts imported plus amounts produced through new investments within the planning horizon plus artificial imports must together at least equal the amount required by the center (in each year). To explain the meaning of the coefficients f_{ikt}, suppose $T = 4$ and let $(f_{ik1}, f_{ik2}, f_{ik3}, f_{ik4}) = (0, 0.35, 0.8, 1)$ for some sector i and some investment project $k \in \mathscr{I}_i$. Then an amount of this investment project designed to permit eventual annual production of one unit of production group i will yield 0 percent of its final production level during year 1, 35 percent during year 2, 80 percent during year 3, and 100 percent during year 4 (and all years after that). That is, the project is planned during year 1, is gradually completed during years 2 and 3, and reaches full operation at the beginning of year 4.

The second set of constraints on sector i is

$$\sum_{k \in \mathscr{R}_i} g_{ijkt} x_{ikt} + \sum_{k \in \mathscr{I}_i} g_{ijkt} x_{ik} \leq z_{ijt} \qquad (j = 1 \ldots n; j \neq i; t = 1 \ldots T). \tag{5.6}$$

g_{ijkt} is the amount of product group j required as input to the production in activity k of one unit of product group i in year t, for $k \in \mathscr{R}_i$. For $k \in \mathscr{I}_i$, g_{ijkt} is the amount of product group j required for investment activity k in year t, comprising both the investment itself (e.g., acquisition of machinery) and the operation of the project for productive purposes. Hence, (5.6) states that total usage in sector i of outputs from other sectors must not exceed amounts assigned.

One further set of constraints on sector i is

$$\sum_{k \in \mathscr{R}_i} h_{ikt} x_{ikt} + \sum_{k \in \mathscr{I}_i} h_{ikt} x_{ik} \leq w_{it} \qquad (t = 1 \ldots T). \tag{5.7}$$

The h_{ikt} are manpower usage coefficients. According to (5.7), total manpower usage in sector i, year t, must not exceed the amount allocated. Additionally, it must of course hold that

$$x_{ikt} \geq 0 \; (k \in \mathscr{R}_i, \mathscr{E}_i, \mathscr{M}_i), \; x_{ik} \geq 0 \; (k \in \mathscr{I}_i), \; x_{iot} \geq 0. \tag{5.8}$$

There may also be "local" restrictions on the x variables, dealing with local plants, and the like. For simplicity, such local restrictions are disregarded here.

The objective of sector i is to maximize foreign exchange earnings:

$$\text{Maximize} \quad \sum_{t=1}^{T} \Bigg(\sum_{k \in \mathscr{R}_i} c_{ikt} x_{ikt} + \sum_{k \in \mathscr{E}_i} c_{ikt} x_{ikt} + \sum_{k \in \mathscr{M}_i} c_{ikt} x_{ikt} + \sum_{k \in \mathscr{I}_i} c_{ikt} x_{ik} + c_{iot} x_{iot} \Bigg). \tag{5.9}$$

$c_{ikt} \leq 0$ for $k \in \mathscr{R}_i$ and $k_i \in \mathscr{I}_i$. $c_{ikt} > 0$ for $k \in \mathscr{E}_i$. $c_{ikt} < 0$ for $k \in \mathscr{M}_i$. Also, c_{iot} is negative with very large absolute value, since the x_{iot} are artificial variables. It is assumed that

$$\max_{k \in \mathscr{E}_i} c_{ikt} < \min_{k \in \mathscr{M}_i} (-c_{ikt});$$

otherwise, unbounded amounts of foreign exchange could be earned by simply importing and re-exporting.

A block-angular LP problem is now formed as follows: The objective functions of the sectors (5.9) are summed over all $i = 1 \ldots n$ to provide the total objective function. Restrictions (5.1), (5.2), and (5.4) are the coupling restrictions. Each sector is restricted by (5.5), (5.6), and (5.7). There may also be further sectoral constraints, relating to local plants for example, but those constraints will not be specified here. Also, all variables must be non-negative [restrictions (5.3) and (5.8)]. This LP problem may, under very pathological conditions, be infeasible (e.g., if the right-hand sides W_t are negative) or have unbounded optimal solutions, but for the present discussion it will be assumed that a finite optimal solution exists. Also, it is assumed that any r_{it}, z_{ijt}, and w_{it} satisfying (5.1), (5.2), (5.3), and (5.4) results in feasible sectoral subproblems. Note, however, that the optimal solution to the LP problem formulated could involve artificial import activities at nonzero levels. In that case, the *LP problem* solution is still feasible, but that solution is not usable for the *planning problem* at hand, meaning that the economic policy figures Q_{it} and W_t are inconsistent with the productive possibilities of the economy.

For solving the LP problem formulated here, a standard LP code could, of course, be used. When the above planning problem was considered in Hungary (around 1962), standard LP codes available in that country could handle problems with at most 100 restrictions (not counting the non-negativity restrictions) (Kornai 1975, p. 374). Solution by standard linear programming was hence not possible, since the problem at hand was too large; instead, a multilevel method had to be used. One obvious candidate was the Dantzig–Wolfe method, but this too, could not be used because of size limitations. The restricted master problem under the Dantzig–Wolfe method would have had as many restrictions as there are constraints of the types (5.1), (5.2), and (5.4), plus one convexity constraint for each sector. As this number of restrictions was apparently also too great, the Kornai–Liptak method was developed. As

pointed out in the earlier discussion of this method (section 3.6), it requires very modest storage capacity for the supremal subproblem.

The block-angular LP problem formulated here does not quite satisfy the assumptions stated at the beginning of section 3.6. Nevertheless, it is not difficult to modify the Kornai–Liptak method, as presented in that section, to enable it to handle the block-angular LP problem formulated here. Some experimental runs of the Kornai–Liptak algorithm, applied to planning problems of this type, were apparently carried out, but details about those problems or about the results are not given (Kornai and Liptak 1965, p. 167; Kornai 1975, pp. 373–375).

5.2.2 THE APPLICATION OF MAN–MACHINE PLANNING TO THE 1966–1970 5-YEAR PLAN

In connection with the preparation of the 1966–1970 Hungarian 5-year plan, another block-angular LP model was constructed (Kornai 1969a; 1969b; 1975, Chapter 28). This model programs production, investment, and foreign trade activities in the final year of the plan (1970) for 491 products. The activities in the model are hence similar to those of the model described in subsection 5.2.1. The 491 products were so-called priority products, actually corresponding to product aggregates (such as "canned meat") rather than fully specified, concrete commodities. Altogether, there were 2,424 activity variables (not counting slack and artificial variables) in the model, and fixing the values of these variables hence determines a particular program for production, investment, and foreign trade relating to the 491 priority products.

The LP model encompasses three hierarchical levels in the national economy. To begin with, there are 45 *sectors*. Each sector has activity variables relating to a group of priority products—the paper or the automobile and tractor industry sectors, for example. These sectors are then grouped into *main branches*. A main branch corresponds either to a ministry or a ministry section. The main branches and sectors constitute the lower levels of the model. Including the center, there are thus three levels.

The structure of the coefficient matrix is displayed in Figure 5.1. The sectoral constraints include capacity constraints in existing plants and export and import constraints on individual products. The main branch constraints include common export constraints for an entire main branch and balance equations that account for the transfer of products between the sectors of a branch (but where the products in question are not transferred to sectors outside the given main branch). The central constraints include manpower restrictions, investment quotas, and balance equations relating to products that are transferred between main branches.

Suppose there are $k(1)$ sectors in main branch 1, $k(2)$ in main branch 2, and so on. Then apparently $\sum_{i=1}^{7} k(i) = 45$; there are 45 sectors altogether. The

parts of the constraint matrix relating to central constraints are denoted $A_0, A_{1,1} \ldots A_{1,k(1)} \ldots A_{7,1} \ldots A_{7,k(7)}$. A_0 does not pertain to any sector but rather to certain activities that are necessary for the whole economy and may be regarded as handled by the center. The subblocks relating to main branch constraints are denoted $B_{1,1} \ldots B_{1,k(1)} \ldots B_{7,1} \ldots B_{7,k(7)}$. The subblocks relating to sectoral constraints are denoted $C_{1,1} \ldots C_{1,k(1)} \ldots C_{7,1} \ldots C_{7,k(7)}$. Denote the variables $x_0, x_{1,1} \ldots x_{1,k(1)} \ldots x_{7,1} \ldots x_{7,k(7)}$. The total problem may then be written as follows:

$$
\begin{aligned}
&\text{Maximize} \quad d_0 x_0 + \sum_{i=1}^{7} \sum_{j=1}^{k(i)} d_{i,j} x_{i,j} \\
&\text{s.t.:} \quad A_0 x_0 + \sum_{i=1}^{7} \sum_{j=1}^{k(i)} A_{i,j} x_{i,j} = a, \qquad \text{(central constraints)} \\
&\qquad \sum_{j=1}^{k(i)} B_{i,j} x_{i,j} = b_i \qquad (i = 1 \ldots 7), \quad \text{(main branch constraints)} \\
&\qquad C_{i,j} x_{i,j} = c_{i,j} \qquad (i = 1 \ldots 7; j = 1 \ldots k(i)), \\
&\qquad\qquad\qquad\qquad\qquad\qquad \text{(sectoral constraints)} \\
&\qquad x_{i,j} \geq 0.
\end{aligned}
\tag{5.10}
$$

Altogether, there are 2,055 constraints (not counting $x_{i,j} \geq 0$). Of these, 67 are central constraints and 90 main branch constraints.

It may be mentioned here that several different objective functions were tried out in the present investigation (e.g., maximization of foreign exchange earnings, minimization of manpower usage). It should also be noted that this model does not cover all of the national economy. For instance, certain sectors of the economy (such as transport) are not included at all.

Problem (5.10) is the economywide, or overall, problem. Because of its three-level character, it may be decomposed into a model system. Suppose the central constraint vector a is partitioned into a_i $(i = 0, 1 \ldots 7)$, $\sum_{i=0}^{7} a_i = a$. Then one obtains the main branch problem (5.11) for each main branch i:

$$
\begin{aligned}
&\text{Maximize} \quad \sum_{j=1}^{k(i)} d_{i,j} x_{i,j} \\
&\text{s.t.:} \quad \sum_{j=1}^{k(i)} A_{i,j} x_{i,j} = a_i, \\
&\qquad \sum_{j=1}^{k(i)} B_{i,j} x_{i,j} = b_i, \\
&\qquad C_{i,j} x_{i,j} = c_{i,j} \qquad (j = 1 \ldots k(i)), \\
&\qquad x_{i,j} \geq 0.
\end{aligned}
\tag{5.11}
$$

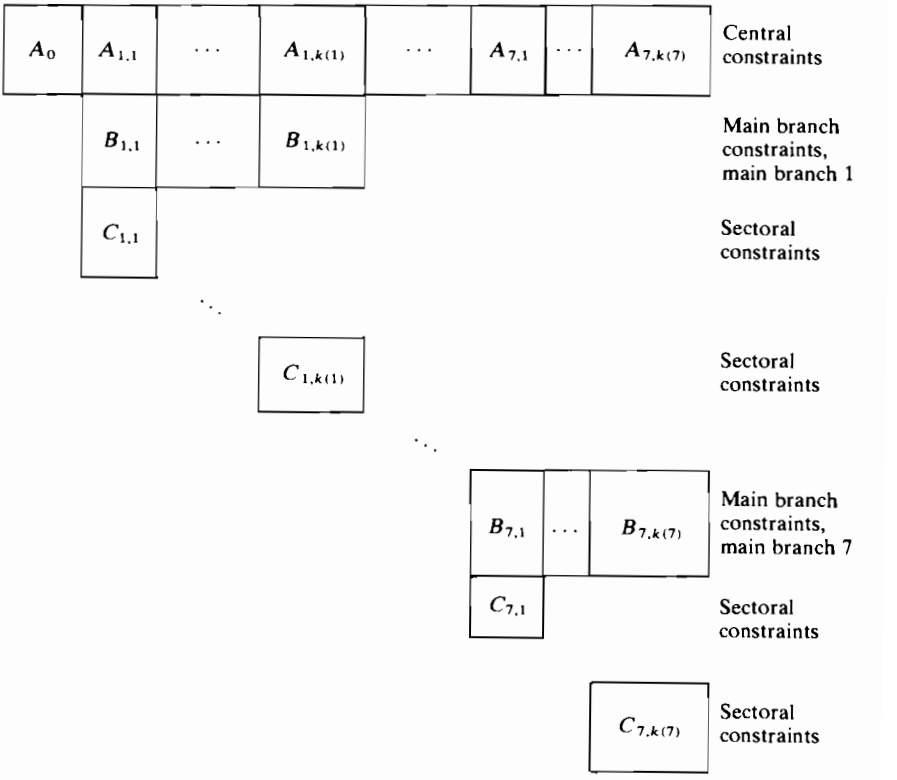


FIGURE 5.1 The coefficient matrix of the Hungarian 5-year plan for 1966–1970.

Suppose, moreover, that a_i and b_i are partitioned into $a_{i,j}$ and $b_{i,j}$ ($j = 1 \ldots k(i)$) such that $\sum_{j=1}^{k(i)} a_{i,j} = a_i$ and $\sum_{j=1}^{k(i)} b_{i,j} = b_i$. Then one obtains the sector problem (5.12) for sector j of branch i:

$$\begin{aligned} &\text{Maximize} && c_{i,j}x_{i,j} \\ &\text{s.t.:} && A_{i,j}x_{i,j} = a_{i,j},\ B_{i,j}x_{i,j} = b_{i,j}, \\ & && C_{i,j}x_{i,j} = c_{i,j},\ x_{i,j} \geq 0. \end{aligned} \tag{5.12}$$

In this fashion, a part, or subproblem, of the overall problem (5.10) can be taken out and tested separately. Conversely, the overall problem (5.10) may be viewed as put together from subblocks, corresponding to main branch and sector problems. Problem (5.10) is hence an example of how multilevel model structures are built from subproblem building blocks.

The overall problem (5.10) could obviously be solved by ordinary LP, in a single-level fashion. This was not possible in the present case, since there was no sufficiently large LP code available in Hungary at the time. Hence, a multilevel method was used. Because the Kornai–Liptak algorithm does not guarantee monotone improvement with each successive iteration, it was decided to use the Dantzig–Wolfe method instead (or, more correctly, a heuristic variant of that algorithm called "man–machine planning"). It was mentioned in the preceding subsection that the restricted master problem of the Dantzig–Wolfe method could also be of considerable size and hence may be outside the reach of available LP codes (indeed, that is why the Kornai–Liptak algorithm was originally developed). However, this difficulty apparently did not arise in connection with the investigation reported here.

In applying the Dantzig–Wolfe method to the economywide problem (5.10), there arises the question of how one should fit the three-level economywide problem into the two-level format of the Dantzig–Wolfe method. At least two principal possibilities present themselves:

1. Only the central constraints are taken as coupling constraints. Each infimal subproblem will then correspond to an entire main branch. That is, the constraints of each infimal subproblem i will be of the form

$$\sum_{j=1}^{k(i)} B_{i,j}x_{i,j} = b_i,\ C_{i,j}x_{i,j} = c_{i,j}(j = 1 \ldots k(i)),\ x_{i,j} \geq 0.$$

There will obviously be seven such infimal subproblems.

2. The central constraints plus all main branch constraints are taken as coupling constraints. Each infimal subproblem will then correspond to a sector. The constraints of each infimal subproblem (i, j) will be simply

$$C_{i,j}x_{i,j} = c_{i,j},\ x_{i,j} \geq 0.$$

There will be 45 infimal subproblems of this type.

Mixtures of these two cases are also possible. That is, the infimal subproblems may correspond to main branches *and* to sectors at the same time.

It is not clear from the sources precisely how the three-level economy-wide problem (5.10) was transformed into a two-level one for the purpose of applying the Dantzig–Wolfe method. Hence, that question will be left aside here, and it will simply be assumed that after a suitable definition of coupling

and local constraints, one obtains the following conventional block-angular LP problem:

$$\begin{aligned}
&\text{Maximize} && c_0x_0 + c_1x_1 + \cdots + c_nx_n \\
&\text{s.t.:} && A_0x_0 + A_1x_1 + \cdots + A_nx_n \leq a, \\
& && B_1x_1 \qquad\qquad\quad \leq b_1, \\
& && \qquad\qquad\quad \ddots \\
& && \qquad\qquad\quad B_nx_n \leq b_n, \\
& && x_0, x_1, \ldots, x_n \qquad \geq 0.
\end{aligned} \tag{5.13}$$

Here, the $x_0, x_1 \ldots x_n$ are variable vectors, and the constant vectors and matrices have suitable dimensions. x_0 denotes activity levels for a set of variables that cannot conveniently be grouped with any of the infimal subproblems. Hence, the corresponding columns—given by A_0—will be added directly at the start as columns to the extremal problem associated with (5.13). Suppose for simplicity that each set $\{x_j | B_jx_j \leq b_j, x_j \geq 0\}$ is bounded. Denote the vector of objective function coefficients for the extremal problem associated with the full set of extreme points w_j and the matrix of coupling constraint coefficients L_j. The extremal problem may hence be written:

$$\begin{aligned}
&\text{Maximize} && c_0x_0 + w_1\lambda_1 + \cdots + w_n\lambda_n \\
&\text{s.t.:} && A_0x_0 + L_1\lambda_1 + \cdots + L_n\lambda_n \leq a, \\
& && e_1\lambda_1 \qquad\qquad\quad = 1, \\
& && \qquad\qquad\quad \ddots \\
& && \qquad\qquad\quad e_n\lambda_n = 1, \\
& && x_0, \lambda_1 \ldots \lambda_n \geq 0.
\end{aligned}$$

λ_j $(j = 1 \ldots n)$ are here variable vectors of suitable dimensions (with as many elements as there are extreme points in $\{x_j | B_jx_j \leq b_j, x_j \geq 0\}$). e_j is a vector with 1 in every position.

As explained above (section 3.3), the Dantzig–Wolfe method starts out with only a few columns of the extremal problem available for each index j (i.e., only a few columns of the matrix L_j). Columns are then generated successively in the adjustment phase. However, to solve the economywide planning problem at hand, the Hungarian team used a heuristic variant of the Dantzig–Wolfe method, as already indicated. This variant differs from the usual Dantzig–Wolfe procedure in some respects.

At the outset, an initial feasible solution to (5.13) was at hand, namely the solution $x_0(1), x_1(1), \ldots, x_n(1)$, taken over from the official national plan, worked out by traditional methods. A second initial feasible solution to (5.13)

could then be obtained by solving the following infimal subproblem, for each $j = 1 \ldots n$:

$$\begin{aligned} &\text{Maximize} && c_j x_j \\ &\text{s.t.:} && A_j x_j \leq A_j x_j(1),\ B_j x_j \leq b_j,\ x_j \geq 0. \end{aligned}$$

This results in an optimal solution $x_j(2)$. Kornai reports that invariably $c_j x_j(2) > c_j x_j(1)$ (Kornai 1969b, p. 213; 1975, p. 609). $x_j(1)$ and $x_j(2)$ can then be used to construct two columns for the restricted master problem.

In each further iteration s, infimal subproblems of the following type are solved for $j = 1 \ldots n$:

$$\begin{aligned} &\text{Maximize} && g_j(s) x_j \\ &\text{s.t.:} && A_j x_j \leq a_j(s),\ B_j x_j \leq b_j,\ x_j \geq 0. \end{aligned} \tag{5.14}$$

Here, the vectors $g_j(s)$ and $a_j(s)$ are determined heuristically on the basis of the solution to the restricted master problem in previous iterations. The optimal solution to (5.14) is used to generate a new column for the restricted master problem. Actually, in one iteration, several different $g_j(s)$ and $a_j(s)$ may be specified, meaning that several different columns are generated in one iteration.

To determine the vectors $g_j(s)$ and $a_j(s)$, intuitive, heuristic methods are used. For instance, if one particular coupling constraint of the restricted master problem is very tight (as evidenced by a high dual variable), then the corresponding components of the $a_j(s)$ vectors (for some or all j) are set "small." In that case, the objective function vector $g_j(s)$ may also be specified so as to minimize the usage of that resource in subproblem j. If, on the other hand, there is positive slack in some coupling constraint of the restricted master problem, the corresponding entries of the $a_j(s)$ are set "large." If, in the jth infimal subproblem, the optimal dual variable associated with some component of $a_j(s-1)$ (i.e., in the previous iteration) is very different from the corresponding dual variable in some other infimal subproblem $i \neq j$, then $a_j(s)$ and $a_i(s)$ may be selected so as to effect a reallocation of that resource between the two infimal subproblems.

The infimal subproblem objective function coefficients $g_j(s)$ and right-hand-side vectors $a_j(s)$ in each iteration are hence not determined automatically by the algorithm itself (as with the usual Dantzig–Wolfe method), but by the researcher, on the basis of his intuition and taking into account the computational results obtained so far. This means that the computations must be interrupted after each iteration, so that the researcher can provide new $g_j(s)$ and $a_j(s)$ for the next iteration. Hence, the method involves a certain interplay of researcher and computer, and for that reason it has been called "man–machine planning" by Kornai himself (Kornai, 1969b). It is interesting to note that a similar type of man–machine planning was used by Beale *et al.* (1965) in

their experimental work with Dantzig–Wolfe decomposition (see section 4.2). The fundamental difference between the ordinary Dantzig–Wolfe method and the heuristic variant described here is that the columns of the restricted master problem of the Dantzig–Wolfe method are derived from extreme points (or extreme rays) of the sets $\{x_j | B_j x_j \leq b_j,\ x_j \geq 0\}$. Not so here—the columns derived from solutions to the subproblems (5.14) will usually be interior points of $\{x_j | B_j x_j \leq b_j,\ x_j \geq 0\}$.

The heuristic method has the following attractive features:

1. It maintains feasibility. That is, if on some iteration a feasible solution to the restricted master problem is obtained, then all later iterations also involve feasible restricted master problems. This means that feasible overall solutions to the original economywide problem can also be recovered immediately.
2. The objective function value of the restricted master problem gets better (or at least, no worse) with each iteration.

It is easily seen that these two properties must hold for the heuristic method, precisely as they hold for the ordinary Dantzig–Wolfe method. The heuristic method was used because the research team involved apparently believed that the ordinary Dantzig–Wolfe method would converge too slowly. The heuristic variant may not necessarily result in an optimal solution to the overall problem (5.10), but that is not very important in the present context. What is required is a "reasonably good" solution after not too many iterations. Indeed, it is even meaningless to talk about "optimum" in the present sort of planning situation, given the uncertainty about some of the data, the arbitrariness of the objective functions and so on.

The heuristic variant of the Dantzig–Wolfe method apparently worked well and produced noticeable improvements over the official national plan in three or four iterations (Kornai 1975, p. 613). The investigation was carried out during 1966–1968. It was of an *experimental* character throughout, meaning that the object was more one of gathering experience in applying formal planning methods than actually producing a definite national plan (Kornai 1969a, p. 135). Nevertheless, the results were discussed with the National Planning Bureau and may have had some effect in shaping the decisions taken.

5.2.3 CONCLUDING REMARKS

At least one additional investigation of some interest from a multilevel point of view has been carried out by Hungarian national economic planners: an LP model related to the 1971–1975 5-year plan was constructed (Kornai 1975, pp. 470–483). This model was in some respects similar to the one described in the preceding subsection. It included production, investment, and foreign trade activities in the final year of the plan (1975). It also had a three-level structure.

The 1971–1975 model was larger than the 1966–1970 one (more constraints and variables), but despite this, it was solved in a single-level fashion, by direct linear programming. This indicates that more powerful standard LP codes had in the meanwhile become available. Nevertheless, the multilevel structure of the model was to some extent taken advantage of, in that separate submodels (corresponding to, for example, single main branches) were tested out and run before the entire model was run.

In concluding this discussion of Hungarian national economic planning, we may ask to what extent the studies surveyed exhibit features of multilevel systems analysis.

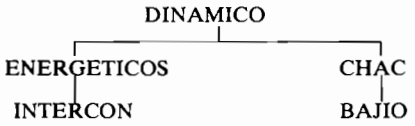
1. All three models (the model of section 5.2.1, the 1966–1970 national economic plan model, and the 1971–1975 model mentioned above) had a multilevel structure (with two or three levels). That is, they had a block-angular LP format. This structure was taken advantage of in that submodels could be tried out separately before the entire model was solved. In this sense, the models may be considered as constructed from a set of building blocks, which implies a certain multilevel quality.
2. The methods used for solution were two-level in the first two cases but not in the last one. This indicates that two-level methods may lose out when more powerful single-level methods become available.
3. Institutionally, both the models themselves and the two-level methods used in the first two studies had multilevel interpretations. That is, the "sectors," "main branches," "center," and so on correspond to actual institutions. Also, the Kornai–Liptak decomposition algorithm and the heuristic variant of the Dantzig–Wolfe decomposition method may be interpreted as iterative dialogues between, for example, the center (the National Planning Bureau) and the main branches (the ministries). In fact, such interpretations appear frequently in Kornai's work. However, the actual solution processes were single-level affairs from the institutional point of view. That is, they were carried out by a single team of researchers (see, for instance, Kornai 1975, p. 479).

5.3 MULTILEVEL NATIONAL ECONOMIC PLANNING IN MEXICO

5.3.1 INTRODUCTION

In the Mexican case study (Goreux and Manne 1973), five different models were constructed. These models concern different levels of detail in the

national economy, and their interrelationships may be represented as follows:



DINAMICO is a macroeconomic planning model covering the time interval 1968–1989. Its objective is to maximize consumption subject to gradual consumption increases over the studied time interval. It divides the economy into 15 sectors, and the productive possibilities of the economy are expressed by an input–output matrix for each model year. It also allows for capacity expansion through investments and for foreign trade. Constraints on labor availability are also included.

ENERGETICOS covers three sectors of the economy: gas and petroleum, electricity, and iron and steel. ENERGETICOS schedules production activities, capacity investments, and foreign trade in these sectors so as to meet exogenous output targets at minimum discounted cost. ENERGETICOS allows for the use of different production technologies, unlike DINAMICO, which is based on an input–output framework and does not permit substitution among alternative production processes. In addition, the product specifications are more detailed in ENERGETICOS than in DINAMICO.

ENERGETICOS does not determine geographical locations of the investments to be undertaken. This problem is handled at the next lower aggregation level, in INTERCON, which is spatially disaggregated. It schedules investments in electricity generation plants and transmission lines in order to meet fixed demands at minimum cost.

CHAC is a model of the agricultural sector. It covers the production of short-cycle agricultural crops, spatially disaggregated into 20 districts. Several different technologies are included for each crop. Labor constraints are included, as are constraints on other resources, such as machinery and irrigation water. Investment activities (e.g., new irrigation canals) and foreign trade activities are also included in the model. The object is to maximize the sum of consumer and producer surpluses. This implies that crop prices are determined endogenously, by the incorporation of step functions for prices.

BAJIO considers only one of the 20 districts of CHAC. It is less aggregated than CHAC, in that production activities on small and large farms are differentiated. Crop prices are fixed exogenously, and the objective function then becomes one of maximizing total producer surpluses.

DINAMICO, CHAC, and BAJIO are LP models. INTERCON is a mixed-integer programming model. ENERGETICOS was solved by linear and mixed-integer programming.

The overall problem of Mexican national economic planning may be stated verbally: Find a national economic development plan that is good for the whole country. However, no explicit overall problem was formalized. Rather, the ensemble DINAMICO-ENERGETICOS-INTERCON-CHAC-BAJIO is taken as a representation of the overall problem, meaning that the researchers involved in this case study proceeded directly to the construction of a multilevel model complex, without first specifying the overall problem (cf. section 2.2.1).

One approach could now be the following: to join all the five models together into one supermodel, i.e., to attempt to construct an explicit overall problem out of the five models. This supermodel could then be solved as a one-shot affair, by ordinary LP or by mixed-integer programming.* Or it could be solved by decomposition. One could imagine that DINAMICO would be used as the coupling constraints of a block-angular LP problem, and that the remaining four models would form local subblocks of that problem. One could then apply some multilevel decomposition algorithm, which would imply an algorithmic, iterative information exchange between the elements of the supermodel, i.e., between the five models DINAMICO, ENERGETICOS, INTERCON, CHAC, and BAJIO.

However, in the Mexican case studies it was not possible to compose such a supermodel. The five models are not compatible in aggregation level, basic model assumptions, and other aspects. For this reason, a different approach was taken. The five models were constructed and solved essentially independently of one another. In particular, there were essentially no iterative information flows between the five models in the solution process. This method of subdividing a total planning task into parts and then solving each part without iterative information exchange and iterative coordination of the parts is called *suboptimization* in Goreux and Manne (1973, p. 3). The Mexican case study may hence be regarded as an exercise in suboptimization.

Actually, there are some information transfers in the Mexican model system, in that certain input parameters in ENERGETICOS and CHAC may be derived from the solution to DINAMICO—in particular, the development of GDP (gross domestic product) and shadow prices on capital and foreign exchange. That is, some *downward* linkages could be established from DINAMICO to the other two models. For instance, the development of GDP was used to aid in the computation of exogenous energy demands for ENERGETICOS. Downward linkages from ENERGETICOS to INTERCON and from CHAC to BAJIO are also possible. For instance, the exogenous agricultural crop prices used in BAJIO may be derived from CHAC (which determines prices endogenously).

In any case, with or without these downward linkages, the Mexican case studies would seem to qualify at best as an extreme, and degenerate, case of multilevel planning, according to the criterion of "multilevel" discussed in section 1.4, since we reserve that term for situations where there are at least

* We disregard here the practical difficulties associated with solving large mixed-integer programming problems.

some rudiments of an iterative information exchange between the subproblems in the model system. For the Mexican work to pass as "true" multilevel planning, one would at least require some one-shot *upward* linkages as well.

The Mexican work is nevertheless of interest from the point of view of multilevel methodology, in that the authors are concerned with the question of whether suboptimization results in a satisfactory solution to the whole planning task (Goreux and Manne 1973, p. 4). In other words, is it a satisfactory methodology to consider each model in isolation, with at most one-shot downward linkages? In order to answer this question, certain experimental upward linkages from ENERGETICOS to DINAMICO and from CHAC to DINAMICO were undertaken. That is, there was in effect a rudimentary and heuristic information exchange. In a later subsection, the upward linkage from ENERGETICOS to DINAMICO will be discussed. As a preliminary to that, however, DINAMICO and ENERGETICOS will be described at somewhat greater length. The upward linkage from CHAC to DINAMICO is similar to the ENERGETICOS–DINAMICO linkage, but more complex in the details.

5.3.2 DINAMICO

DINAMICO is a highly aggregated macroeconomic planning model. It is of the LP type, with 316 constraint rows and 421 activity columns (variables). The variables relate to

1. Production outputs and capacity increases in 15 sectors of the economy
2. Usage, upgrading, and downgrading of five labor force skill classes
3. Exports, foreign capital inflows, and remittances to foreign countries
4. Macroeconomic quantities (GDP, gross investment, gross savings, consumption)

The Mexican economy is divided into 15 sectors (agriculture, mining, various industrial sectors, construction, commerce, transportation, services) that together account for the entire national product. For each sector, there is only one technology available, reconstructed from tables of historical interindustry transactions (i.e., historical input–output tables). That is, substitution between alternative technologies is not allowed. Production outputs and investments are measured in billions of 1960 pesos, not in natural units.

The five skill categories of labor range from "unskilled agricultural workers" to "engineers and scientists." The activities mentioned under (2) include education to upgrade labor from a lower category to a higher one and migration of agricultural laborers to one of the nonagricultural categories.

DINAMICO is a dynamic model, in that the workings of the economy are studied at 3-year intervals, from 1968 to 1989 (where 1968 is involved only in setting initial conditions and 1989 in setting terminal conditions). The

objective of the model is to maximize aggregate consumption in 1971. This objective, however, in a certain sense implies consumption maximization over the entire planning period 1971–1989, since one of the constraints enforces a "gradualist consumption path": increments in aggregate consumption are required to grow 7 percent per year.

The individual restrictions of the model are as follows:

Material balances. These state that the net output in sector j ($j = 1 \ldots 15$) must be at least equal to all uses of that output: for consumption, investments in various sectors, exports, and inputs into production in other sectors. Since there are 15 sectors, and since the years 1971, 1974, 1977, 1980, 1983 and 1986 (6 years) are considered, there are obviously 90 constraints of this type. Consumption use of a product group is determined as aggregate consumption times the average propensity to consume that product group. This means that final consumption demands for the output of each sector are constrained to vary in fixed proportions.

Capacity constraints. Total output in sector j, in each of the years considered, must not exceed a certain base-year output plus capacity increments made available through investment activities. There are also certain restrictions on terminal investments (in 1989), designed to avoid so-called horizon effects.

Labor demands and supplies. There are equations defining the total demand for labor in each of the years 1971, 1974, 1977, 1980, 1983 and 1986, as a function of output activities in the 15 sectors. Total usage of labor in each skill category, in any one of those years, must be less than or equal to exogenously projected available amounts plus amounts made available through skill upgrading (educational) activities and skill substitution activities.

Foreign trade. There are constraints that define export earnings and foreign exchange deficit (or surplus) and also define how this deficit is to be financed. These constraints are simple definitional equations. There are also certain (inequality) restrictions on the inflow of private capital from abroad (designed to avoid excessive foreign ownership of Mexico's capital stock).

Macroeconomic definitions and constraints. For each of the considered years, GDP, gross domestic savings, and gross domestic investment are defined through conventional macroeconomic identities (equations). Additionally, there is a constraint on the savings increase: The savings increase must be less than or equal to the marginal propensity to save times the increase in GDP.

Gradualist consumption path. This restriction (equality) has alredy been mentioned above.

Additionally, upper and lower bounds are imposed on some individual activities, mainly export activities. These bounds are not included in the constraint count of 316.

DINAMICO is solved by ordinary (single-level) LP. The optimal solution provides output and investment targets for the 15 different sectors in the different model years. It also provides projections of GDP and foreign exchange transactions. Furthermore, there are some dual variables which may be of interest. For instance, one may obtain shadow prices on foreign exchange (Goreux and Manne 1973, pp. 139–144). Such shadow prices are of interest from a multilevel point of view, since they can be used as input parameters in lower-level models, as was mentioned in the previous subsection.

5.3.3 ENERGETICOS

ENERGETICOS is a model of the energy sectors of the Mexican economy. The energy sectors are here taken as the petroleum and gas industry, the electricity industry, and the iron and steel industry. The last is included because it is a significant user of energy, and process choices in that industry may affect the development of the energy sectors in general. The overall objective of ENERGETICOS is to choose investments in alternative processes in the three sectors so as to meet output requirements at a minimal discounted cost, taking into account intersectoral flows. This means that many of the variables of the problem refer to concrete investment projects in the three sectors, and what is desired is a cost-minimal set of such projects to be undertaken. The precise *location* of the selected investments is not determined by ENERGETICOS—that is, it contains no regional dimension. The initial year covered by the model is 1974. Material balances are included for each year in the period 1974–1980. Additionally, three 5-year intervals are considered in the electricity sector: 1981–1985, 1986–1990, and 1991–1995.

ENERGETICOS has a block-angular structure. The constraint coefficient matrix may be displayed as in Figure 5.2. This block-angular structure was not taken advantage of in the final runs of the model. However, it was utilized in that the

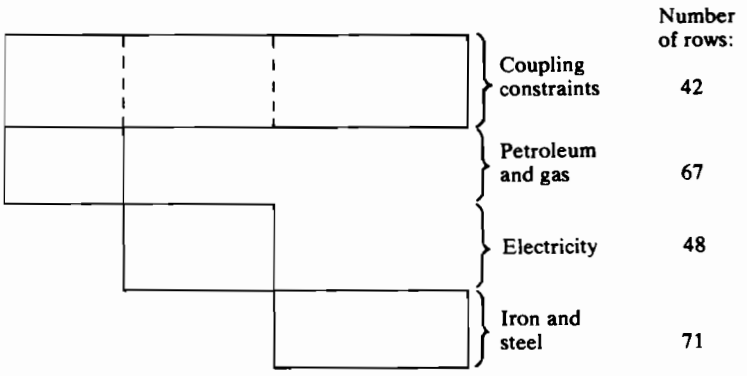


FIGURE 5.2 The coefficient matrix of ENERGETICOS.

three submodels, corresponding to the three sectoral subblocks, could be tested out separately.

The *coupling constraints* are of two kinds:

1. *Material balances* of industrial fuel, electrical peak power, and electrical energy.* There is one such balance for each of the three, and for each year considered (7 years for industrial fuel, and 10 years for electrical peak power and electrical energy; the difference is accounted for by the three 5-year intervals that are considered for the electricity sector). There is a triangular order among the three sectors in that industrial fuel is produced in the petroleum and gas sector but utilized in the other two. The petroleum and gas sector is assumed not to use peak power or electrical energy. The electricity sector uses industrial fuel but produces peak power and electrical energy. The iron and steel sector uses all three commodities. Hence, there is a one-way flow of commodities from petroleum and gas to electricity to iron and steel. Each industrial fuel balance (for a given year) states that the amount produced in old and new installations in the gas and petroleum sector minus amounts used in the electricity sector and in new installations in the iron and steel sector must be at least equal to the exogenous demand. Industrial fuel consists of residual fuel oil and natural gas. Together they are counted as one product, measured in caloric equivalents. The peak power and electrical energy balances state that amounts produced in old and new installations in the electricity industry minus amounts utilized in new installations in the iron and steel industry must be at least equal to exogenous demands.

2. *Cost definition equations*, which convert future costs into present values. A 10 percent annual discount rate is used. Cost elements associated with the various activities include depreciation and interest on investment projects undertaken and variable costs associated with running the plants acquired. These variable costs are, for instance, costs of industrial fuel produced within Mexico and imported, labor costs, and costs of the raw materials required by the iron and steel sector (iron ore, metal scrap, and so on). Some of these items may be imported, such as metal scrap and some part of the crude oil required by the petroleum and gas sector. For imports, the exchange rate 12.5 pesos per U.S. dollar is used (apparently the conventional rate at the time of the investigation). However, imports are charged with an additional 15 percent of their costs, representing a certain amount of domestic protection.

The *local constraints* of the three sectors have the following contents:

1. *The petroleum and gas sector*: There are rows expressing the input and output of different petroleum products associated with existing refinery

* An electricity plant produces joint products: power and energy. Power is the output at an instant of time. Energy is the integral of power output over time. Power is measured in kW, energy in kWh.

capacity and the different refinery processes under consideration for investments. Those processes are of three kinds: "conventional," "visbreaking," and "maximum hydrogene processing." The only investment activities are refinery activities. In particular, undertaking investments in opening additional gas or oil fields is not considered within the model. The inputs in the petroleum and gas sector are gas and crude oil. The availability of gas and crude oil from domestic sources is projected exogenously, for the different years considered in the model. Additional amounts may be imported. There are also output requirements for various petroleum and gas products; the output requirements in different years for industrial fuel (gas and residual oil) are given in the corresponding rows of the coupling constraint block. Those requirements are dependent on the particular investment projects in the electricity and iron and steel sectors. For other petroleum products, such as gasoline, kerosene, and diesel oil, the output requirements restrict only the gas and petroleum sector, and hence these requirements enter on the right-hand side in the gas and petroleum subblock.

In summary, then, the main activities in the gas and petroleum sector subblock are refining in existing and new refineries, and imports of gas and crude oil. The restrictions express limits on domestic availability of gas and crude oil, and output requirements on certain refined products, which are independent of activities in the electricity and iron and steel industries. Output requirements for industrial fuel are expressed through the coupling constraints.

2. *The electricity sector.* The investment activities available to the electricity sector are new fossil fuel and nuclear electricity generation plants. Only residual oil and natural gas are considered as fossil fuels. The local constraints of the electricity sector include requirements for expansion of transmission capacity as additional generators are installed. There are also equations defining the requirement of industrial fuel in the different model years (these requirements then enter into the coupling constraints pertaining to industrial fuel). There are also local constraints dealing with so-called *service shifting*, meaning that newer, more economical plants will be operated at full capacity, whereas older plants will be utilized to absorb the fluctuations in the daily load curve. The outputs of the electricity sector, peak power and energy, enter into the coupling constraints of ENERGETICOS.

3. *The iron and steel sector.* This sector is simplified in that steel ingots (not rolled products) are regarded as the end product. For each model year, the demand for steel ingots in the economy is calculated (exogenously). The existing capacity is deducted, and the difference must be covered by new investments in the sector. There are restrictions to this effect in the local constraint block. Additionally, there are equations defining the requirements for iron ore, scrap and coke. The requirements of the iron and steel sector for industrial fuel, peak power, and electrical energy enter in the coupling constraints. There are six types of investment choices available, each a different

kind of integrated process for manufacturing steel ingots (e.g., "prereduced pellets + blast furnace + LD converter").

ENERGETICOS was solved in a one-level fashion, as an ordinary LP problem. However, since the investment projects considered are really indivisible, it should ideally be solved as a mixed integer problem. After some simplification, this was also done (utilizing a variant of the Benders algorithm). In the discussion of linkages between DINAMICO and ENERGETICOS in the next subsection, we will be concerned only with the LP solution.

A solution to ENERGETICOS may be summarized in the form of a time-phased vector of investments in the different types of available projects in the three sectors and of foreign exchange expenditures associated with that investment plan. The basic variant solution was obtained by setting the discount rate to 10 percent and applying a 15 percent import protection cost, as indicated above.

5.3.4 LINKAGES BETWEEN DINAMICO AND ENERGETICOS

As mentioned above, the Mexican case study is concerned with *suboptimization*. Suboptimization comes about when an overall problem is factored into subproblems that are then solved independently, without any iterations of information exchange between the subproblems (Goreux and Manne 1973, p. 3). It is relevant to ask whether this procedure of suboptimization is acceptable. Or, in other words, would the subproblem solutions have changed drastically if there had been some iterative information exchange with accompanying reoptimizations? If the answer to this second question is no, then suboptimization is acceptable. The following experiments in linking DINAMICO and ENERGETICOS are designed to shed light on this question of the acceptability of suboptimization.

ENERGETICOS was constructed and solved largely independently of DINAMICO. In one respect, though, ENERGETICOS takes certain input data from DINAMICO: The exogenous demands for industrial fuel and other refined petroleum products, electrical peak power and energy, and steel ingots were checked against the results of DINAMICO. That is, DINAMICO provides production targets for the 15 sectors of the Mexican economy at different points in time, and those targets may be of help in deriving the exogenous delivery requirements incorporated in ENERGETICOS. ENERGETICOS may hence be characterized as a quantity-taker with respect to deliveries to other sectors of the economy. It is a price-taker with respect to resources used as inputs (e.g., foreign exchange, labor, capital equipment, metal scrap, crude oil, natural gas).

In order to conclude that suboptimization is acceptable in the present case, i.e., where DINAMICO and ENERGETICOS are formulated and solved essentially independently of one another, one would like to assure oneself, first, that the exogenous demands for deliveries from the energy sectors to other sectors of

the economy do not change as a consequence of the solution produced by ENERGETICOS; and, second, that the prices on inputs utilized in ENERGETICOS do not change drastically as a consequence of the solution produced by ENERGETICOS.

Consider now the first of these two points. As mentioned earlier, a solution to ENERGETICOS is a time-phased vector of investments in different projects in the three sectors and of foreign exchange expenditures. By varying two input parameters, a series of six different solutions was obtained. Those two parameters are the discount rate (which was set at 10 percent and 20 percent) and the foreign exchange scarcity premium (which was set at 0 percent, 30 percent, and 60 percent). The foreign exchange scarcity premium is not the same as the 15 percent protection against imports mentioned earlier; the 15 percent protection refers only to import costs, whereas the foreign exchange scarcity premium is an additional charge, levied on *net* imports (i.e., the difference between import costs at conventional prices and export earnings). The basic ENERGETICOS case, referred to in the end of section 5.3.3, was obtained by fixing the discount rate at 10 percent and the foreign exchange scarcity premium at 0 percent, that is, no exchange scarcity premium on top of the 15 percent import protection. By considering all the other combinations of the two parameter values used, one apparently obtains six different cases.

As ENERGETICOS is run with the six parameter combinations, the choices of investment projects in the three sectors vary from case to case. Total solution cost also varies from one case to another. For purposes of comparison, that cost was measured in the same way in all six cases, by setting the discount rate at 10 percent and the exchange scarcity premium at 0 percent. That is, the six solutions were *generated* by considering different parameter combinations, but for *evaluating* those solutions, the same parameter combination (the base-case combination) was used for all six solutions. It then turned out that the difference in total cost between the most and least expensive solutions was only about 4 percent, despite the fact that these two solutions involve different investment programs. If one now assumes that the long-run price elasticity of demand in the rest of the economy for products from the three sectors of ENERGETICOS is at most unity, it follows that process substitution within the three sectors (i.e., choice among the six different solutions) could not lead to more than a 4 percent change in exogenous demands for the output products of the ENERGETICOS sectors. Hence, under a fairly wide range of solutions to ENERGETICOS, outside demands, using information derived from the DINAMICO solution, remain fairly constant; that is, the particular solution chosen for ENERGETICOS does not affect outside demand drastically (Goreux and Manne 1973, pp. 282–285).

Consider now the question of whether the prices on inputs utilized by the three ENERGETICOS sectors might not change as a consequence of the solution

produced by ENERGETICOS. This was investigated by the following methodology: The six different ENERGETICOS solutions were incorporated as activity columns in DINAMICO. DINAMICO was then rerun, and the degree of difference between the dual variable values of DINAMICO and those of the DINAMICO variant without the ENERGETICOS columns incorporated was checked. The dual variables of DINAMICO may be interpreted as a price system of sorts for some of the resources in the Mexican economy (such as labor of different classes and foreign exchange). The prices on inputs used in ENERGETICOS were not taken directly from the DINAMICO dual solution. This would not have been entirely possible, anyway, since the two models have rather different coverage (e.g., DINAMICO does not give shadow prices on metal scrap, which is one of the input commodities in ENERGETICOS). Nevertheless, if the dual prices of DINAMICO change drastically as a consequence of the incorporation of ENERGETICOS solutions, then one may conclude that it is probably not justifiable to solve DINAMICO and ENERGETICOS independently.

Consider now the question of how to incorporate the six ENERGETICOS solutions in DINAMICO. Naively, one might suggest that the six ENERGETICOS solutions could be used directly to form activity columns for DINAMICO in somewhat the same way as columns are formed for the restricted master problem under the Dantzig–Wolfe decomposition method. This, however, is not possible, since the two models are largely incompatible: their aggregation levels and coverage differ too much. For instance, DINAMICO uses only one output for each product sector, measured in pesos, whereas ENERGETICOS has two kinds of outputs for the electricity sector (peak power and energy, measured in physical units), and several different kinds of outputs for the petroleum and gas sector. On the other hand, labor is disaggregated into skill classes in DINAMICO, with separate constraints for the different classes, whereas labor enters only indirectly in ENERGETICOS (as parts of the cost coefficients). Nevertheless, the ENERGETICOS solutions do require the usage of some of the resources for which there are constraints in DINAMICO: most important among these are capital and foreign exchange.

DINAMICO is based on 3-year intervals from 1968 onward; ENERGETICOS is based on 1-year intervals from 1974 to 1980 and 5-year intervals after that. Now consider the years 1974, 1977, and 1980. These years are covered in both models. Let F_{it} be the requirement of foreign exchange in year t ($t = 1$: 1974; $t = 2$: 1977; $t = 3$: 1980) associated with ENERGETICOS solution i ($i = 1 \ldots 6$). Let K_{it} be the requirement of investment capital of that solution in year t. Suppose that one adds the following terms to the foreign exchange equations of DINAMICO, for $t = 1, 2, 3$:

$$\sum_{i=1}^{6} (\lambda_i + \mu_i) F_{it}.$$

In the same fashion, suppose one adds the following terms to the investment

rows of DINAMICO for $t = 1, 2, 3$:

$$\sum_{i=1}^{6} (\lambda_i + \mu_i) K_{it}.$$

Moreover, one adds the constraints

$$\sum_{i=1}^{6} \lambda = 1, \sum_{i=1}^{6} \mu_i = -1, \lambda_i \geq 0, \mu_i \leq 0.$$

This modification of DINAMICO allows for some substitution (without modification, there are no substitution possibilities in the model). By setting $\lambda_i = 1$ and $\mu_i = -1$ for some fixed i, one apparently obtains no substitution at all. That is, the foreign exchange and capital usage in DINAMICO remains unchanged. By setting $\lambda_i = 1$ and $\mu_j = -1$ for $i \neq j$, one allows for ENERGETICOS solution j to be completely replaced by ENERGETICOS solution i. By considering the whole set of λ_i and μ_i satisfying the constraints $\sum_{i=1}^{6} \lambda_i = 1, \lambda_i \geq 0, \sum_{i=1}^{6} \mu_i = -1, \mu_i \leq 0$, one allows for a whole range of modifications of the foreign exchange and capital usage in DINAMICO, where those modifications derive from alternative solutions to ENERGETICOS.

DINAMICO was rerun after being modified in the way described here, and it was discovered that the dual variables associated with the six restrictions relating to foreign exchange and investment in 1974, 1977, and 1980 hardly changed at all. From this, one may infer that the solutions to ENERGETICOS do not noticeably affect the shadow prices associated with DINAMICO (Goreux and Manne 1973, pp. 285–289).

The upshot of this is that the suboptimization methodology utilized in the Mexican study—i.e., solving DINAMICO and ENERGETICOS independently, without any information exchange or mutual readjustments—is an acceptable one.

5.3.5 CONCLUSIONS AND COMPARISON WITH MULTILEVEL NATIONAL ECONOMIC PLANNING IN HUNGARY

We may now ask: To what extent are aspects of multilevel systems analysis represented in the Mexican case study? We note first two somewhat superficial points:

1. ENERGETICOS has a block-angular structure, and the same is true for CHAC. These structures were utilized in that separate parts of those models could be tested out independently before the entire model was run. (A similar procedure was also followed in the Hungarian work; see section 5.2.3.) However, the block-angular structures were not utilized in the final runs. Instead, ordinary (single-level) LP was used.

2. In connection with the Mexican case study, certain experimental computations with the Dantzig–Wolfe method were undertaken (Kutcher 1973). These experiments have already been mentioned in Chapter 4. They have little to do with national economic planning *per se.* Also, ENERGETICOS and INTERCON were solved as mixed-integer programming problems using a variant of the Benders algorithm (Goreux and Manne 1973, Chapter V.I).

More fundamentally, the Mexican work represents an extreme case of multilevel planning: suboptimization, i.e., where a given overall problem is factored into smaller ones that are then solved independently, without mutual readjustments and coordination based on an iterative information exchange. It is nevertheless interesting from a multilevel methodological point of view in that the authors have considered the question: Does suboptimization produce acceptable results? To investigate this question, some linkage experiments were performed. The linkages established had to be of a somewhat heuristic nature. The models involved (e.g., DINAMICO and ENERGETICOS) are so different in scope, formulation, and other aspects that an automatic linkage, like the information transfer between the restricted master problem and infimal subproblems of the Dantzig–Wolfe method, was not possible. The Mexican work illustrates how heuristic linkages between fairly incompatible models in a model system can be established and how these linkages can aid in answering the question of whether suboptimization is acceptable.

One may also compare the Hungarian and Mexican case studies in national economic planning. In both instances, there is a "natural" three-level hierarchy involved. In the Hungarian work (the 1966–1970 and 1971–1975 5-year plans), there are three levels represented: center, main branch, and sector. Similarly, the Mexican work may be said to comprise models on three levels. DINAMICO may be taken as a "center model." CHAC and ENERGETICOS are sectoral models, and INTERCON and BAJIO are models of parts of sectors. It is evident from the Hungarian and Mexican work that there are two different approaches to national planning.

The first approach is to build a decomposable LP supermodel with a three-level block-angular structure. Under this approach, one may well start by constructing submodels, or subblocks, pertaining to, for example, individual sectors, but in the end all the pieces are put together to one large model. The important thing is that all the pieces are compatible and fit together easily. This is the approach taken in the Hungarian work. Under this approach, one then has the choice of solving the resulting supermodel by some multilevel decomposition algorithm, implying an algorithmic, iterative exchange of information between the different subproblems, or by ordinary, single-level LP. Both choices are exemplified in the Hungarian work. The 1966–1970 5-year plan model was solved by "man–machine planning" (a heuristic variant of Dantzig–Wolfe decomposition), and the 1971–1975 5-year plan model was solved by ordinary LP.

The second approach is to construct a model system where the models on different levels are independent of, and incompatible with, one another. This is the Mexican approach. Under this approach, the total model system cannot be solved as a supermodel by (for instance) ordinary LP. A formal iterative procedure of the decomposition type also cannot be applied, because of model incompatibilities. One is then forced to suboptimize, possibly with some informal model linkages added.

Suboptimization may very well involve some optimality loss, compared with the first approach (the Hungarian one). For that reason, the Hungarian approach may be preferable to the Mexican one. Nevertheless, there are some reasons that one may prefer (or be forced) to use suboptimization: There may be such a wide discrepancy in the statistical data base between sectors that it is impossible to construct a totally compatible model system. Second, the suboptimization approach corresponds to institutional realities. In the Mexican case study, the different models in the model system were constructed by different research teams. In a more routine planning situation, different planning agencies (e.g., ministries) tend to their separate branch or sector problems.

Third, the suboptimization approach allows for separate (incompatible) modeling treatment in each model to take into account particular sectoral or regional conditions. This may not be possible if uniform modeling across the economy is necessitated by complete model compatibility.

5.4 A PROBLEM OF REGIONAL PLANNING

5.4.1 THE DEVELOPMENT NETWORK

The focus now shifts from national to regional planning. This section draws on Alekseev (1975, pp. 63–101), and Alekseev *et al.* (1974, pp. 81–103).* We will follow the development in Alekseev (1975) in first presenting two subproblems. We then indicate how those subproblems can be put together to an overall problem, and we then outline and discuss the two-level solution method proposed by Alekseev.

Consider the problem of developing a geographical region of a country. Such development may be considered a complex project and can be represented by a PERT-type network, for example as shown in Figure 5.3. In this case, a forest area is to be felled, and a railway, a harbor, a power station, and a sawmill are to be constructed. The network displays the order in which the various project parts must be completed.

In general, let the regional development network be denoted by G. The nodes (or events) are indexed by i or j. The arcs are denoted (ij). Any $(ij) \in G$ signifies a particular activity that is to be completed as part of the overall

* Alekseev and Kozlov (1977) is a short English-language summary.

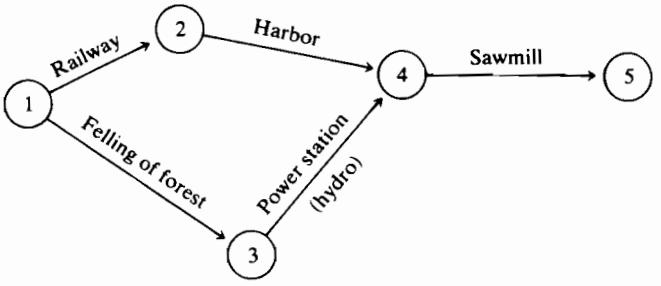


FIGURE 5.3 A development network.

project. Activity (ij) takes τ_{ij} time periods (integer months) to complete. Let t_{ij} denote the time period that activity (ij) is started. However, activity (ij) cannot be started until all activities earlier than (ij) in the network have been completed, that is, not until event i has been reached. Let that time period be denoted t_i.

In carrying out the activities, certain resources (e.g., cement, asphalt) are utilized. Let there be K such resources altogether. Suppose t is any period. Let $Q_{ij}^k(t - t_{ij} + 1)$ be the usage of resource k in activity (ij) in period t if that activity is started in period t_{ij} $(1 \leq t - t_{ij} + 1 \leq \tau_{ij})$. That is, the resource usage depends only on the number of periods ago that the activity was started. Let u_{tk} be the price of resource k in period t. The total cost of activity (ij), as a function of t_{ij} (the start period), is hence

$$\sum_{t=t_{ij}}^{t_{ij}+\tau_{ij}-1} \sum_{k=1}^{K} Q_{ij}^k(t - t_{ij} + 1)u_{tk}.$$

It is desired to schedule all activities so that the total sum of resource costs (over time) is minimized. This can be written as:

$$\text{Minimize with respect to } t_{ij} \text{ and } t_j\text{:} \quad \sum_{(ij)\in G} \sum_{t=t_{ij}}^{t_{ij}+\tau_{ij}-1} \sum_{k=1}^{K} Q_{ij}^k(t - t_{ij} + 1)u_{tk}. \tag{5.15}$$

There are the following restrictions to observe:

$$t_j = \max_{\substack{i \\ (ij)\in G}} (t_{ij} + \tau_{ij}), \text{ for all events } j \in G. \tag{5.16}$$

Equations (5.16) means that an event j is attained only when all activities leading up to that event have been completed.

$$t_{ij} \geq t_i, \text{ for all activities } (ij) \in G. \tag{5.17}$$

This means that activity (ij) cannot be started before event i is attained.

$$t_{ij} + \tau_{ij} \leq T, \text{ for all activities } (ij) \in G. \tag{5.18}$$

According to (5.18), the whole network must be completed no later than the beginning of some predetermined period T. Additionally, one has the definition:

$$t_1 = 1. \tag{5.19}$$

Finally, the t_i and t_{ij} must be integers.

Expressions (5.15)–(5.19) define a combinatorial decision problem. The decision variables are apparently the t_{ij} and t_j. Note that the τ_{ij} are constants. This means that each activity takes a specified number of periods, calculated in advance. The associated resource usage is also calculated in advance. This implies that there is only *one* technology for completing each activity. Note also that each activity is supposed to be carried out in consecutive periods. That is, if a particular activity (ij) takes two periods ($\tau_{ij} = 2$), then one could not, for instance, start the activity in period 11, let it rest in periods 12 and 13, and then complete it in period 14.

Problems (5.15)–(5.19) can be solved by dynamic programming methods in simple cases; for more complicated problems, an approximate method can be used (Alekseev 1975, pp. 71–75; Alekseev *et al.* 1974, pp. 93–95). We will not be concerned here with the details of that solution method, but assume only that an optimal solution to (5.15)–(5.19) can be obtained.

5.4.2 AN LP MODEL FOR RESOURCE PRODUCTION

Now assume that the K different resources needed in the development network are produced in a set of factories. Let the production activities in these factories be denoted summarily by the vector x. Let a_{tk} be a vector that transforms the activity vector x into output of resource k in period t ($k = 1 \ldots K$, $t = 1 \ldots T-1$). That is, $a_{tk}x$ is the output of resource k in period t of the total production complex (set of factories). Let the cost vector be c. For a given solution to the network problem (5.15)–(5.19), one may compute total resource demands Q_{tk}, for each period t and resource k. It is now desired to produce these resources as cheaply as possible. This may be written as:

$$\begin{aligned} &\text{Minimize} && cx \\ &\text{s.t.:} && a_{tk}x \geq Q_{tk} \ (t = 1 \ldots T-1; k = 1 \ldots K), \\ & && x \in X, \end{aligned} \tag{5.20}$$

where X is defined by some set of linear inequalities (including $x \geq 0$) and expresses constraints on production capacities in individual factories, etc. It is assumed that (5.20) is feasible for any set of Q_{tk} resulting from a solution to (5.15)–(5.19).

5.4.3 THE OVERALL PROBLEM AND A TWO-LEVEL SOLUTION METHOD

Alekseev (1975) does not explicitly formulate an overall problem. However, it is easy to see that the two subproblems formulated in the preceding two subsections may be combined into the following overall problem:

$$\begin{aligned}
&\text{Minimize} \quad cx \\
&\text{s.t.:} \quad -a_{tk}x + Q_{tk} \leq 0 \; (t = 1 \ldots T-1, k = 1 \ldots K), \\
&\qquad x \in X, \\
&\qquad Q_{tk} = \sum_{(ij)\in G} Q_{ij}^{k}(t - t_{ij} + 1) \quad (t = 1 \ldots T-1, k = 1 \ldots K), \qquad (5.21) \\
&\qquad t_j = \max_{\substack{i \\ (ij)\in G}} (t_{ij} + \tau_{ij}), \text{ for all events } j \in G, \qquad (5.22) \\
&\qquad t_{ij} \geq t_i, \text{ for all activities } (ij) \in G, \qquad (5.23) \\
&\qquad t_{ij} + \tau_{ij} \leq T, \text{ for all activities } (ij) \in G, \qquad (5.24) \\
&\qquad t_1 = 1. \qquad (5.25)
\end{aligned}$$

We may rewrite this overall problem as

$$\begin{aligned}
&\text{Minimize} \quad cx \\
&\text{s.t.:} \quad -a_{tk}x + Q_{tk} \leq 0 \; (t = 1 \ldots T-1, k = 1 \ldots K), \\
&\qquad x \in X, \\
&\qquad (Q_{11} \ldots Q_{1K}, Q_{21} \ldots Q_{2K} \ldots Q_{T-1,1} \ldots Q_{T-1,K}) \in Q,
\end{aligned} \qquad (5.26)$$

where Q is a set defined by the restrictions (5.21)–(5.25). Q is apparently a discrete set of points, and we note in passing that (5.26) is of a form suitable for an application of the Benders decomposition algorithm. However, another two-level method, a heuristic one, was used to solve the overall problem (5.26) in Alekseev (1975); this will be discussed below.

Let (5.15)–(5.19) be the supremal subproblem and (5.20) the infimal subproblem. That is, there is only one infimal subproblem. For a given solution to the supremal subproblem, resource demands Q_{tk} may be computed by means of the relation (5.21). Suppose those demands are inserted as the right-hand side of the infimal subproblem (5.20). When (5.20) is solved with that right-hand side, a set of resource prices u'_{tk} (dual variables) is obtained. These resource prices can then be inserted into the objective function of the supremal subproblem, to generate a new supremal subproblem solution (i.e., a new development schedule).

In other words, one iteration of the adjustment phase could be carried out as follows: The supremal subproblem (5.15)–(5.19) is solved, taking as resource

costs u_{tk} the dual variables associated with the infimal subproblem in the previous iteration. This results in a set of resource demands, which are transferred to the infimal subproblem. The infimal subproblem is solved, and the dual multipliers associated with the constraints $a_{tk}x \geq Q_{tk}$ are transferred to the supremal subproblem. The supremal subproblem objective function is then revised, taking those dual variables as resource costs.

Actually, if one merely inserts the new dual multipliers u'_{tk} into the objective function of the supremal subproblem directly, then the iterative process may exhibit sharp oscillations. For that reason, the objective function coefficients for the supremal subproblem have to be modified. Let u^s_{tk} be the dual multipliers associated with the infimal subproblem in iteration s. Let $\hat{u}^s_{tk}$ be the objective function coefficients utilized in the objective function (5.15) of the supremal subproblem of the same iteration. Then $\hat{u}^{s+1}_{tk}$ is formed as follows:

$$\hat{u}^{s+1}_{tk} = (1-\alpha_s)\hat{u}^s_{tk} + \alpha_s u^s_{tk}$$

where α_s is chosen such that $\alpha_s \to 0$ as $s \to \infty$, $\sum_{s=1}^{\infty} \alpha_s \to \infty$. (One series of weights α_s satisfying these conditions is $\alpha_s = 1/s$.)

On each iteration, the value of the solution to the overall problem arrived at is given by the solution to the infimal subproblem, with the right-hand sides Q_{tk} taken from the supremal subproblem solution in the same iteration. Let that solution value be denoted z_s. z_s does not necessarily decrease with each iteration (Alekseev 1975, pp. 88–90). For a stopping rule, z_s and z_{s-1} may be compared. If $|z_s - z_{s-1}| < \varepsilon$ (some predetermined positive constant), the process stops.

To start the first iteration, resource prices may be arbitrarily specified. The adjustment phase goes on for some (limited) number of iterations. In the execution phase, a solution to the original problem is obtained from the supremal and infimal subproblem. That is, the supremal subproblem provides the development schedule, and the infimal subproblem the associated resource production plans.

An overall problem of the type considered here was formulated for the development of the Boguchany territorial industrial complex in Siberia. It was solved by the method outlined here. For that purpose a computer program was written that can handle networks with up to 500 activities and LP subproblems with up to 100 restrictions (Alekseev 1975, p. 90). For the case of the Boguchany territorial industrial complex, a total of four iterations was sufficient to attain a satisfactory solution (Alekseev *et al.* 1974, p. 98).

5.4.4 DISCUSSION OF THE TWO-LEVEL METHOD FOR REGIONAL PLANNING

No justification for the above two-level method is given in Alekseev (1975) or Alekseev *et al.* (1974). However, a certain justification does exist. Suppose $\hat{u}^s_{tk}$ $(t = 1 \ldots T-1, k = 1 \ldots K)$ are objective function coefficients for the

supremal subproblem in iteration s of the adjustment phase. Let Q_{tk}^s be the resulting resource demands, given by the solution to the supremal subproblem. Let x^s be an optimal solution to the infimal subproblem, and let u_{tk}^s be dual multipliers associated with the resource demands Q_{tk}^s. If now $u_{tk}^s = \hat{u}_{tk}^s$, for all t and k, then $(x^s, Q_{11}^s \ldots Q_{1K}^s, Q_{21}^s \ldots Q_{2K}^s \ldots Q_{T-1,1}^s \ldots Q_{T-1,K}^s)$ is an optimal solution for the overall problem (5.26). This follows from Everett's theorem,* since $(x^s, \ldots Q_{tk}^s \ldots)$ minimizes

$$cx + \sum_{t=1}^{T-1} \sum_{k=1}^{K} u_{tk}^s(-a_{tk}x) + \sum_{t=1}^{T-1} \sum_{k=1}^{K} \hat{u}_{tk}^s Q_{tk}$$

$$= cx + \sum_{t=1}^{T-1} \sum_{k=1}^{K} \hat{u}_{tk}^s(-a_{tk}x) + \sum_{t=1}^{T-1} \sum_{k=1}^{K} \hat{u}_{tk}^s Q_{tk}$$

over $X \times Q$, and since $-a_{tk}x^s + Q_{tk}^s \leq 0$, with strict inequality implying that the corresponding u_{tk}^s is zero.

Hence, $u_{tk}^s = \hat{u}_{tk}^s (t = 1 \ldots T-1, k = 1 \ldots K)$ represents a sufficient optimality condition. The adjustment phase of the two-level method may be interpreted as a search for resource prices satisfying the sufficient optimality condition. Because of the averaging procedure used in calculating $\hat{u}_{tk}^{s+1}$ (see section 5.4.3), price oscillations are dampened, but the $\hat{u}_{tk}^s$ may converge to a set of resource prices that do not satisfy the sufficient optimality condition (as can be shown by counterexamples). Also, the Q_{tk}^s may not converge, but oscillate, despite the averaging procedure. Furthermore, the solution values z_s need not decrease monotonically, as mentioned earlier.

For the reasons just mentioned, the two-level method for regional planning must be regarded as heuristic. Together with the two earlier case studies described in this chapter, the present study exemplifies heuristic multilevel approaches.

REFERENCES

Alekseev, A. M. 1975. Mnogourovnevye sistemy planirovaniia promyshlennogo proizvodstva. (Multilevel Systems for Planning Industrial Production, in Russian.) Novosibirsk: Nauka (Siberian department).

* *Everett's theorem.* Suppose $\bar{x}$ is an optimal solution to the problem

$$\begin{aligned} &\text{Minimize} && f(x) + ug(x) \\ &\text{s.t.:} && x \in X, \end{aligned}$$

where $u \geq 0$, and u and $g(x)$ have m components. Then $\bar{x}$ is an optimal solution also for the problem

$$\begin{aligned} &\text{Minimize} && f(x) \\ &\text{s.t.:} && g_i(x) \leq y_i (i = 1 \ldots m), x \in X, \end{aligned}$$

where $y_i = g_i(\bar{x})$ if $u_i > 0$, and $y_i \geq g_i(\bar{x})$ if $u_i = 0$. (Lasdon 1970, pp. 402–403.)

Alekseev, A. M., and L. A. Kozlov. 1977. Optimization of a long-term program for the formation of a territorial-production complex, pp. 301–311. In H. Knop (ed.), The Bratsk–Ilimsk Territorial Production Complex. CP-77-3. Laxenburg, Austria: International Institute for Applied Systems Analysis.

Alekseev, A. M., L. A. Kozlov, and V. N. Kriuchkov. 1974. Setevye modely v perspektivnom planirovanii razvitiia proizvodstva. (Network Models in Perspective Planning of the Development of Production, in Russian.) Novosibirsk: Nauka (Siberian department).

Baranov, E. F., V. I. Danilov-Danil'ian, and M. G. Zavel'skii. 1971. On a system of optimal perspective planning. (In Russian.) Ekonomika i matematicheskie metody 7:332–350.

Beale, E. M. L., P. A. B. Hughes, and R. E. Small. 1965. Experiences in using a decomposition program. Computer Journal 8:13–18.

Ellman, M. 1973. Planning Problems in the USSR. Cambridge: Cambridge University Press.

Fedorenko, N. P. 1974. Optimal Functioning System for a Socialist Economy. Moscow: Progress Publishers.

Fedorenko, N. P. (ed.). 1975. Sistema modelei optimal'nogo planirovaniia. (A System of Models for Optimal Planning, in Russian.) Moscow: Nauka.

Ganczer, S. 1973. The main features of applying mathematical methods in planning in Hungary. Jahrbuch der Wirtschaft Osteuropas 4:43–66.

Goreux, L. M., and A. S. Manne (ed.). 1973. Multi-level Planning: Case Studies in Mexico. Amsterdam: North-Holland.

Kantorovich, L. V. 1976. Mathematics in economics: Achievements, difficulties, perspectives. Mathematical Programming 11:204–211.

Katsenelinboigen, A. I., and E. Iu. Faerman. 1967. Centralism and economic independence in the socialist economy. (In Russian.) Ekonomika i matematicheskie metody 3:331–346.

Kornai, J. 1965. Mathematical programming as a tool in drawing up the five-year economic plan. Economics of Planning 5(3):3–18.

Kornai, J. 1969a. Multilevel programming—A first report on the model and on the experimental computations. European Economic Review 1(1):134–191.

Kornai, J. 1969b. Man–Machine planning. Economics of Planning 9(3):209–234.

Kornai, J. 1975. Mathematical Planning of Structural Decisions. 2nd ed. Amsterdam: North-Holland.

Kornai, J., and T. Liptak. 1965. Two-level planning. Econometrica 33:141–169.

Kronsjö, T. 1963. Iterative pricing for planning foreign trade. Economics of Planning 3(1):1–22.

Kutcher, G. P. 1973. On decomposing price-endogenous models, pp. 499–519. In L. M. Goreux and A. S. Manne (ed.), Multi-Level Planning: Case Studies in Mexico. Amsterdam: North-Holland.

Lasdon, L. S. 1970. Optimization Theory for Large Systems, New York: Macmillan.

Martynov, G. V., and A. K. Pitelin. 1969. Experimental investigations of the approximating scheme of multi-stage optimization. (In Russian.) Ekonomika i matematicheskie metody 5:526–540.

Pugachev, V. F. 1974. Problems of multi-stage optimization of national economy, pp. 477–483. In J. Los and M. W. Los (ed.), Mathematical Models in Economics. Amsterdam: North-Holland.

Pugachev, V. F., G. V. Martynov, V. G. Mednitskii, and A. K. Pitelin. 1972. Multi-stage optimization with a local criterion of a general type. (In Russian.) Ekonomika i matematicheskie metody 8:635–649.

Pugachev, V. F., G. V. Martynov, V. G. Mednitskii, and A. K. Pitelin. 1973. Multi-stage optimization with concrete forms of the local criterion. (In Russian.) Ekonomika i matematicheskie metody 9:204–217.

Trzeciakowski, W. 1973. Economic calculus and the system of foreign trade management in a centrally planned economy. Jahrbuch der Wirtschaft Osteuropas 4:115–137.

Zauberman, A. 1975. The Mathematical Revolution in Soviet Economics. London: Oxford University Press.

6 Planning of Production and Sales Programs in Corporations

6.1 INTRODUCTION

6.1.1 THE PLANNING PROBLEM

In this chapter we consider multilevel methods for solving a class of problems relating to production and sales planning in business firms. Abstractly, the planning problem is formulated as a block-angular LP problem:

$$
\begin{aligned}
\text{Maximize} \quad & c_1x_1 + c_2x_2 + \cdots + c_nx_n \\
\text{s.t.:} \quad & A_1x_1 + A_2x_2 + \cdots + A_nx_n \leq a, \\
& B_1x_1 \qquad\qquad\qquad\quad \leq b_1, \\
& \qquad B_2x_2 \qquad\qquad\quad \leq b_2, \\
& \qquad\qquad\quad \ddots \\
& \qquad\qquad\qquad\quad B_nx_n \leq b_n, \\
& x_1, x_2 \ldots x_n \geq 0.
\end{aligned}
\tag{6.1}
$$

The usual interpretation of (6.1) in this connection is the following: A firm consists of a headquarters group on the first level and n departments (or divisions) on the second.* Each department controls a set of activity variables. For department j, x_j is the vector of activity levels for the activities pertaining to that department. Those activities may relate to purchases of raw materials, production, deliveries to other departments, and sales to outside customers.

* "Division" usually denotes a subunit in a divisionally organized corporation, "department" a subunit in a functionally organized corporation (Jennergren 1975, pp. 11–16). In this chapter, we use the term department, because the two corporations in the following case studies are functionally organized.

Each department is limited by some local constraints, $B_j x_j \leq b_j$. These may, for instance, express local capacity constraints in the physical plant of department j, or upper and lower bounds on sales to outside customers of that particular department.

Additionally, the constraints $A_1x_1 + A_2x_2 + \cdots + A_nx_n \leq a$ limit the choice of activities by all departments taken together. They can express, for example, the usage of certain joint resources, such as machine capacity, raw materials, and manpower. Also, the coupling constraints may express balance conditions on the transfer of raw materials, semifinished products, and so on from one department to others. The restrictions $A_1x_1 + A_2x_2 + \cdots + A_nx_n \leq a$ are often referred to as corporate restrictions, since they affect all departments, or the whole corporation. In contrast, the restrictions $B_j x_j \leq b_j$ affect only department j and may therefore be referred to as departmental.

The planning problem (6.1) is a fairly short-run one. That is, it aims to select a production and sales program for some coming time period, like 1 to 3 months. Obviously, the *detailed* scheduling of individual jobs through the factory is not considered in (6.1). The goal of the firm in this planning situation is taken as one of maximizing contribution to profit. The vectors c_j express contributions to profit associated with the different activities. This goal of maximizing contribution to profit may be considered appropriate for a short-run planning situation like the one considered here.

Problem (6.1) is an abstract statement of the overall problem of this chapter. It is obviously a block-angular LP problem. Nonlinear formulations of the total planning problem facing the corporation have also been proposed, though (see, e.g., Kulikowski 1975). Before discussing multilevel methods for solving the overall problem (6.1), we may inquire whether (6.1) is a "real" problem. Do planning problems of type (6.1) actually exist in real companies? The answer is yes. At least in some companies, planning problems like (6.1) do arise in connection with production and sales planning. The two case studies described in this chapter utilize planning problem data derived from two real companies and hence pertain to planning situations which do exist. We may thus conclude that the planning object—production and sales planning problems of type (6.1)—does in fact exist. Hence it is meaningful to discuss solution strategies for that planning object.

6.1.2 PLANNING PROCEDURES BASED ON DECOMPOSITION METHODS

Suppose now that some company faces the planning problem (6.1). The simplest way to solve it, i.e., to arrive at a production and sales plan, may well be to assemble all information about (6.1) (the complete problem description) in one place, at headquarters, and then solve (6.1) directly, for instance by ordinary single-level LP.

However, in some cases it may not be feasible to assemble all information about (6.1) in one place. That is, information is dispersed among different organizational subunits of the corporation. For instance, it is quite natural to assume that department j has some knowledge of the constraints $B_j x_j \leq b_j$, since they refer to, for example, the physical plant of that department. Moreover, it could also be that department j is unable, or even unwilling, to supply headquarters with a precise description of its departmental constraints $B_j x_j \leq b_j$. However, each department can usually be expected to be able to answer questions of the following type: "What would your production and/or sales plans be under the following conditions . . . ?" Where those conditions refer, for instance, to a specific set of transfer prices (Jennergren 1971a, pp. 11–12; Polterovich 1972, p. 444).

In situations such as this, it seems natural to attempt to construct a planning procedure founded on some decomposition method, since the overall problem (6.1) is a block-angular one and hence suited for an application of, e.g., the Dantzig–Wolfe method, and since the utilization of such a planning procedure would usually not violate the condition that information about (6.1) is dispersed among various subunits of the corporation. In effect, what is being proposed is to use a two-level planning procedure, founded on some decomposition method, where messages are actually exchanged between different subunits in the corporation during the adjustment phase, and where each subunit iteratively performs certain calculations. That is, we are talking about a category 3 situation, in the classification of section 2.2.2. When a decomposition method is used as a purely computational tool, information is exchanged between different subblocks of a computer program, which are called upon to solve different subproblems. In a category 3 situation, information is exchanged between different organizational subunits, and each subunit has its own subproblem to solve at different points in the planning process.

It has been pointed out time and again that planning procedures founded on decomposition methods could be used in departmentalized (or divisionalized) corporations. When the Dantzig–Wolfe decomposition method was published, it was almost immediately pointed out that it has a certain resemblance to budgeting, or planning, procedures in real business firms (Almon 1963, Baumol and Fabian 1964). It was then suggested that it could also actually be implemented as a planning procedure in companies. The same suggestion has been put forth for several other decomposition algorithms. The literature on the usage of planning procedures founded on decomposition methods is by now very large, as evidenced by the fact that quite a few surveys have been published (among them, Atkins 1974, Bailey 1976, Burton and Obel 1977, Ennuste 1972, Freeland 1973, Jennergren 1971a, Martines-Soler 1974, Polterovich 1969, Ruefli 1974).

It is customary to divide planning procedures for solving problems like (6.1) founded on decomposition methods into two groups: price-directive and resource-directive. These labels derive from the nature of the information exchange between the supremal and infimal subproblems in the adjustment phase, where the supremal subproblem is considered to "belong to" headquarters, and the infimal subproblems to departments. Under a price-directive approach, the information going from headquarters to departments in each iteration of the adjustment phase is a tentative price vector, associated with the corporate constraints. The information going back from departments to headquarters includes tentative quantities. These quantities result from activity decisions that would be taken by the departments if they could carry out transactions in jointly utilized scarce resources and intermediate products at the prices announced by headquarters. The Dantzig–Wolfe method is apparently a price-directive one. So is the Lagrangean method (section 3.7). That method, however, has only been discussed for certain nonlinear overall problems and hence cannot immediately be used as the basis for a planning procedure for the overall problem (6.1), at least not without some adaptation. Quite a few price-directive two-level planning methods have been proposed in the literature, for example, Baumol and Fabian (1964), Charnes *et al.* (1967), Hass (1968), Jennergren (1972, 1973) Kydland (1975), and Mandel' (1973).

Under resource-directive approaches, the information going from headquarters to departments in each iteration of the adjustment phase is a tentative partition of the right-hand side of the corporate constraints among departments. Such a partition may be viewed as tentative quantities of various semifinished products and jointly utilized resources. The messages going back to headquarters include information about how departmental payoffs would change in response to changes in the tentative quantities. The ten Kate and Kornai–Liptak methods are resource-directive ones.* A fair number of resource-directive two-level planning methods have also appeared (for instance, Burton *et al.* 1974, Freeland and Baker 1975, Jennergren 1971b, ten Kate 1972, Kornai and Liptak 1965, Pervozvanskaia and Pervozvanskii 1966, and Zschau 1967).

The literature on the usage of decomposition methods as the basis for planning tools in corporations is hence very large. It is, however, almost entirely of a theoretical nature. It is a most disappointing fact, as also mentioned by Ruefli (1974, p. 361) that so far no implementations of planning methods founded on decomposition methods have been attempted in real corporations (as far as is known). However, a few simulation studies of the performance of decomposition methods as planning tools have been

* The ten Kate decomposition method is a special case of the Benders algorithm applied to block-angular LP problems, as pointed out in section 3.5.4. In this chapter, we use the label ten Kate rather than Benders, to conform with the literature.

undertaken. Jennergren and Müller (1973) is one such study, using small, randomly generated planning problems. This chapter presents two other simulation studies, one pertaining to a paperboard manufacturer (Ljung and Selmer 1975), and the other to a slaughterhouse (Christensen and Obel 1976). As indicated earlier, these two case studies utilize real planning problem data taken from the two companies and then simulate the production and sales planning process using the Dantzig–Wolfe and ten Kate decomposition methods. That is, what would the consequences have been *if* the decomposition method had been applied as the basis for a decision-making system in the given company? What would the information transfers between headquarters and departments, and vice versa, be like? What subproblems would the different organizational subunits solve in each iteration, and what would the resulting solution be like?

In the literature, one can find lists of desirable properties that decomposition methods should possess, if they are to be used as planning instruments in real corporations (e.g., Jennergren 1971a, pp. 23–28; Malinvaud 1967). For instance, they should be well defined in the sense that it is completely clear what each organizational subunit is supposed to do at each point of the planning process, what information is to be exchanged, and so on. The most important property, however, is probably that a "good" solution to the overall planning problem (6.1) should be obtained with only a very small number of iterations of information exchange in the adjustment phase. In a real company, not many iterations of some planning scheme calling for subproblem solving and information exchange between different organizational subunits will be undertaken, probably three or four at most. The following case studies address themselves to the issue of whether a good solution to the planning problem can be obtained in a small number of iterations of information exchange.*

In a sense, the following two case studies are concerned with the efficiency of decomposition algorithms. Nevertheless, the emphasis is here very different from that in Chapter 4 on computational experiences with Dantzig–Wolfe decomposition for LP problems, where the main interest is in the total *machine time usage* to obtain an *optimal* (or near-optimal solution). If 100 iterations of information exchange between the supremal and infimal subproblems are necessary, that does not matter, since a very large number of iterations may quite easily be carried out by a computer. Here, we are concerned with the quality of the solution obtained *after only a few iterations.* To further avoid misunderstandings, it is only that ultimate solution that is to be implemented in actual production and sales activities in the company. The various tentative plans computed by headquarters and departments in the iterative information

* There is no contradiction between formulating (6.1) as an optimization problem and then attempting to obtain a "good" solution through the use of some decomposition method as a planning tool. That is, one strives towards optimization but recognizes that in the end one has to settle for a good solution.

exchange of the adjustment phase are *not* to be implemented. They are only trial plans on the way to the final and definitive one.

It should be mentioned that the emphasis here is different from that in Chapter 5 on national economic planning, too. The overall planning problem (6.1) is in its economic meaning very similar to a national economic planning problem involving a planning agency and economy sectors of the type considered in Kornai (1975). Both problems involve planning physical production for some future time period. (In fact, a few of the references cited here refer to national economic planning rather than to planning in industrial firms.) It is really only the scale of the problem that differs. Yet, there is again the difference in the solution methodology, in that it is suggested here that the planning problem (6.1) is to be solved through an institutional arrangement that involves assigning different subproblems to different organizational subunits and then performing an iterative information exchange between these subunits. This solution methodology was not used by Kornai and his associates. Rather, the 1966–1970 5-year planning problem was solved by one single group of investigators, within one organization, using a heuristic variant of Dantzig–Wolfe decomposition as a numerical tool for problem solving on one single computer.

6.2 A SIMULATION STUDY OF A PLANNING PROCEDURE BASED ON THE DANTZIG–WOLFE METHOD IN A PAPERBOARD FACTORY

This case study is based on the work of Ljung and Selmer (1975). The company involved is a Swedish manufacturer of a variety of wood products; this study concerns only part of the company's activities, the manufacture of paperboard. Paperboard is produced from raw materials delivered from other units within the company. However, in what follows we will speak of the paperboard factory as "the company," for simplicity disregarding the fact that this factory is actually in itself a division of a larger corporation.

6.2.1 THE PLANNING PROBLEM OF THE PAPERBOARD FACTORY

The planning problem to be described concerns the production and sales of paperboard. The typical situation at the time of the investigation was that demand exceeded the supply possibilities of the company. This meant that it was important to coordinate sales and production plans. The planning problem discussed below refers to a 1-year period. The data for the problem were taken from a larger corporate planning model covering both paper pulp and paperboard. Once that part of the model that refers to paper board was taken out, it had to be transformed in certain ways (e.g., some variables and coefficients had

to be redefined). In the end, a block-angular LP problem with coefficient matrix like that displayed in Figure 6.1 was obtained.

As can be seen, there are four subblocks. Each refers to a sales area, like "Scandinavia." The activities refer to volumes of sales to individual customers. There are about 200 different qualities of paperboard (i.e., 200 different products) that can be manufactured in the factory. However, each customer usually buys one particular product mix. This means that each customer can be represented by a single activity variable. In a few cases, customers require varying product mixes and so must be represented by more than one activity variable. The total number of variables is hence somewhat larger than the total number of customers. Altogether, there are 652 variables. Of these, 137 refer to sales area 1, 22 to sales area 2, 272 to sales area 3, and 221 to sales area 4. The departmental constraints are of a rather special nature, as also indicated in Figure 6.1. They consist exclusively of upper and lower bounds on sales to individual customers. This may appear strange at first but may actually arise naturally in a short-run planning situation where one may have contracted to deliver certain minimum quantities, is forced to allocate one's deliveries among customers due to excessive demands, etc.

One may wonder why a planning problem with the particular block-angular structure depicted in Figure 6.1 was formulated. In particular, why are there four subblocks (and not, for instance, six)? And why does the second subblock only encompass 22 sales variables, whereas the third has 272 such variables? The answer is that this company is grouped into four sales areas, and the division of customers between sales areas in the formulation above corresponds to organizational realities.

There are only six corporate constraints. Four of these refer to annual capacities in four different machine groups (for board making, rolling, plastic coating, and sheet cutting). The remaining two restrictions state certain

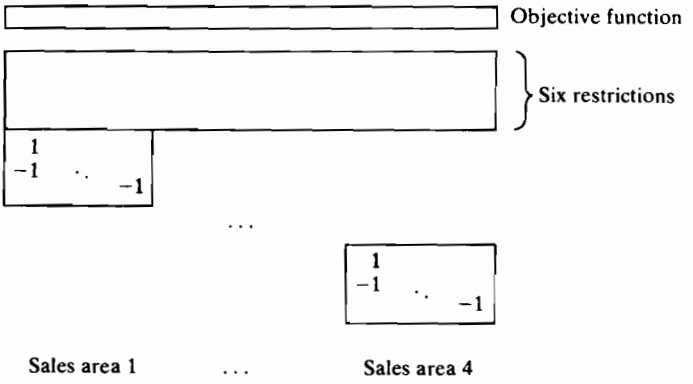


FIGURE 6.1 Coefficient matrix for the paperboard factory planning problem.

requirements on the product mix (certain percentages of the simpler qualities are necessary for cleaning the board-making machines). It may be remarked that there are no constraints on raw materials (paper pulp). The company has its own paper pulp supply, as indicated earlier. This supply is always sufficient for paperboard manufacturing.

The objective function coefficients express contribution to profit associated with each customer. They are computed starting with the sales prices from which certain items are deducted: the opportunity cost of paper pulp (which can be sold directly to outsiders), the cost of certain other raw materials (such as plastic), the cost of electricity and steam, and handling and transportation costs.

Implied in the above overall planning problem formulation is a certain division of planning labor between headquarters and departments (sales areas). Headquarters will coordinate the sales plans for the sales areas in such a manner that the physical production constraints (the six corporate constraints) are satisfied. This means, in particular, that headquarters is associated to some extent with the production function. A somewhat different division of labor is implied by a problem formulation with the coefficient matrix exhibited in Figure 6.2. In this case, the production function is taken as a separate department of its own. The departmental constraints of the production department correspond to the six corporate ones from the earlier formulation. The decision variables of the production department refer to different product qualities (not individual customers). The corporate constraints in this second formulation become balance constraints: the supply from the production department must at least equal demands in the sales areas. There will hence be about 200 such balance equations. The role of headquarters is now one of balancing the delivery plans of the production department with the sales plans of the sales areas. This is a somewhat more limited task than in the first overall problem formulation. (Note that we are here comparing two *different overall*

FIGURE 6.2 Coefficient matrix for an alternative formulation of the paperboard factory planning problem.

problem formulations, not two ways of constructing a two-level subproblem hierarchy relating to the same overall problem.)

Ljung and Selmer considered the second problem formulation as well, but they do not use it in their simulation experiments. The reason is that the second formulation is a more complex one. The application of Dantzig–Wolfe decomposition to planning problems of the kind considered here involves (among other things) finding realistic prices for the corporate constraints. In the first formulation above, only six such prices are involved (one for each of the four machine groups, plus one for each of the two product mix constraints). In the second formulation, about 200 transfer prices are involved (for transfers of the 200 different product qualities from the production department to the sales areas). This illustrates the difference in complexity between the two formulations.

6.2.2 INFORMATION DISPERSAL AND INFORMATION FLOWS

The economic interpretation of the Dantzig–Wolfe method has been discussed above (section 3.3.6). That interpretation coincides, in fact, with the way it would be implemented as a planning method in a corporation. At each iteration of the adjustment phase, headquarters calculates a tentative price vector p^t, associated with the coupling constraints of the supremal subproblem (the restricted master problem) of that iteration t. Each department j then solves the infimal subproblem

$$\begin{aligned} &\text{Maximize} && c_j x_j - p^t A_j x_j \\ &\text{s.t.:} && B_j x_j \leq b_j,\ x_j \geq 0. \end{aligned}$$

This calculation may be interpreted as an attempt to determine departmental activity levels that maximize departmental profit (the difference between $c_j x_j$ and $p^t A_j x_j$), under the assumption that jointly utilized resources and semifinished products can be freely traded at the prices p^t. Suppose x_j^t is an optimal solution. Two pieces of information from department j are then added to the supremal subproblem in the next iteration: the column $(A_j x_j^t)$, which may be interpreted as a quantity proposal from department j; and $(c_j x_j^t)$, the contribution to profit for the whole corporation resulting from the divisional activity levels given by x_j^t.

The precise form of the messages from headquarters to department j and from department j back to headquarters depends on the precise way in which information about the overall planning problem (6.1) is dispersed in the corporation. In the case of the paperboard company, it is most natural to assume that each department j knows its departmental constraints $B_j x_j \leq b_j$. Since each department is a sales area, it is also logical to assume that it knows c_j (i.e., the contributions to profit associated with the different activities).

Headquarters knows the corporate constraints $\sum A_j x_j \leq a$ (both the right-hand side and the left-hand-side coefficients). In this situation, the information going from headquarters to department j in iteration t of the adjustment phase is the vector $(p^t A_j)$ (not merely p^t). After receiving this message, department j is capable of constructing its infimal subproblem, since the other data for that subproblem (c_j, B_j, and b_j) are known by department j in advance. The information going back from department j is $(c_j x_j')$ and x_j'. With that information, headquarters can construct the new column $(A_j x_j')$ for the supremal subproblem.

If a different initial dispersal of information about (6.1) is assumed, somewhat different information flows result. If a planning procedure based on Dantzig–Wolfe decomposition is to be used, then headquarters must *at least* know a, the right-hand side of the corporate constraints. Department j must *at least* know $B_j x_j \leq b_j$, its own departmental constraints. The remaining pieces of information necessary for constructing the infimal and supremal subproblems can be transferred during the iterative information exchange.

6.2.3 THE SIMULATION EXPERIMENT

The production and sales planning problem of the paperboard manufacturing company was described in section 6.2.1. The performance of the Dantzig–Wolfe method as a planning tool, for reaching a decision on that planning problem, has been simulated. It may first be noted that the departmental subproblems at each iteration of the adjustment phase are very simple in this case: the only restrictions are upper and lower bounds (for each variable). Hence, an optimal solution is obviously to set each variable equal to the lower or upper bound, depending on whether the corresponding objective function coefficient is negative or not.

In order to obtain a good solution to the planning problem in a small number of iterations, it is necessary to obtain a feasible solution to the restricted master problem as quickly as possible—in the first or second iteration. This means that the very first set of proposals obtained from departments should preferably result in a feasible solution to the supremal subproblem. The reason this is preferable is obvious: as soon as a feasible supremal subproblem has been obtained, a feasible solution to the original planning problem can be recovered at any point later in the planning process. Whether the first set of proposals satisfies the feasibility requirement depends on the initial prices announced by headquarters that are used to generate those proposals. In other words, the problem of immediately obtaining a feasible solution to the restricted master problem can largely be reduced to one of selecting a good set of initial prices associated with the corporate constraints.

In the present case, there are six prices to be specified initially, one for each corporate restriction. The first four of these prices refer to scarce resources

TABLE 6.1 Solution Value of Production and Sales Plan, after Different Iterations of Information Exchange

Iteration	Value (in % of True Optimum)	Best Upper Bound (in % of True Optimum)
1	77.65	105.9
2	86.81	105.9
3	96.93	101.0
4	98.00	101.0

(machine time capacities), and the remaining two to product mix constraints. In order to attempt to ensure that the machine time capacities were not exceeded, the initial prices associated with the machine time constraints were set "high." The remaining two were set to zero. It turned out that this set of prices did result in a feasible restricted master problem in the first iteration.*

The further progress of the planning procedure is shown in Table 6.1. It should be noted that the solution value for each iteration refers to the value of the restricted master problem at that iteration. The convergence of the restricted master problem solution value is seen to be quite rapid. After three iterations, almost 97 percent of the maximum total has been attained, and the gap between actual solution value and upper bound is 4 percent. (It may be remarked that a total of nine iterations was required to attain the true optimum.) Some additional runs were performed with modified right-hand sides (i.e., minor modifications in the planning problem data), with similar results (rapid convergence during the initial iterations).

6.2.4 IMPLEMENTATION OF THE PLAN

After terminating the iterative information exchange (i.e., after terminating the adjustment phase), a decision must be made about implementation. As was pointed out earlier, the various tentative plans proposed by departments or the successive restricted master problem solutions are not implemented one after the other. It is only at the end of the planning process, in the execution phase, that one particular decision is implemented. There are several methods for that implementation. (See the discussions in sections 3.3.6 and 4.2 of forms of the execution phase under the Dantzig–Wolfe decomposition method.)

In the first place, headquarters can announce the weights given by the last solution to the restricted master problem and instruct departments to combine their previous proposals in accordance with those weights and then implement the resulting weighted activity vector in actual production and sales activities.

* The iteration count is as follows: The initial tentative price vector is announced in iteration 0. The supremal subproblem is then solved for the first time at the start of iteration 1.

In that case, the resulting solution value is the same as that of the restricted master problem in the final iteration.

In the second place, headquarters can allocate the right-hand side of the corporate constraints to departments and then instruct them to find production and sales plans on their own, taking into account the allocations made. That is, suppose L_j is a matrix constructed from plan proposals submitted by department j during the iterative planning process. Let the vectors $\bar{\lambda}_1, \bar{\lambda}_2, \ldots, \bar{\lambda}_n$ denote an optimal solution to the restricted master problem in the last iteration of the adjustment phase. Then headquarters can instruct each department j to formulate and solve the following subproblem:

$$\begin{aligned} &\text{Maximize} && c_j x_j \\ &\text{s.t.:} && A_j x_j \leq L_j \bar{\lambda}_j,\ B_j x_j \leq b_j,\ x_j \geq 0. \end{aligned}$$

The resulting solution is implemented in actual production and sales over the coming planning period. This produces a feasible solution to the overall planning problem (assuming $\bar{\lambda}_1, \bar{\lambda}_2, \ldots, \bar{\lambda}_n$ is feasible for the restricted master problem). The resulting total payoff is no lower than the solution value of the last restricted master problem but may well be higher.

A third implementation method is for headquarters to announce the dual prices associated with the corporate constraints of the restricted master problem in the last iteration of the adjustment phase. Let those prices be given by the price vector $\bar{p}$. Each department is then instructed to solve the subproblem

$$\begin{aligned} &\text{Maximize} && c_j x_j - \bar{p} A_j x_j \\ &\text{s.t.:} && B_j x_j \leq b_j,\ x_j \geq 0, \end{aligned}$$

and implement the resulting solution over the coming planning period. This implementation method often results in infeasible solutions to the original planning problem.*

All three implementation methods were investigated in the present case. Table 6.2 states what the resulting solution value would be, if the planning process had been halted after one to four iterations, for each of the three implementation methods. It is seen from Table 6.2 that when weights are announced, the resulting solution value is the same as that of the restricted master problem in the last iteration. Right-hand-side allocations result in higher total payoff than weights (production orders). Prices result in infeasible solutions to the original planning problem.

* This is so, even if $\bar{p}$ is an optimal price vector associated with the corporate constraints of the original problem. See section 2.1.2 on coordinable and noncoordinable two-level subproblem hierarchies relative to an overall problem of block-angular LP type.

TABLE 6.2 Value of the Solution to the Original Planning Problem, for Different Implementation Methods in the Execution Phase (values in percent of true optimum)

Iteration	Weights Announced (Production Orders)	Right-Hand-Side Allocations	Prices
1	77.65	[a]	[a]
2	86.81	[a]	[a]
3	96.93	98.46	102.00[b]
4	98.00	99.63	98.96[b]

[a] Not stated in Ljung and Selmer 1975
[b] Infeasible solution

6.2.5 SOME CONCLUSIONS

The results of this simulation study of the Dantzig–Wolfe method as a planning instrument are quite positive: for a real-world planning problem, it was possible to reach a good solution in no more than three or four iterations. Moreover, it was easy to generate an initial feasible solution to the restricted master problem. As for implementing the production and sales plan, the best implementation method was right-hand-side allocations.

An obvious question at this point is whether these fairly positive results are due to specific features of the particular planning problem. Two such features are the small number of corporate constraints and the structure of the subblocks. Obviously, one would expect a better plan (higher solution value) in, say, three iterations if the number of corporate constraints is small rather than large. In that respect, the features of the problem situation may have been influential. We will return to this question, the role of the number of corporate constraints, in the next section. The structure of the subblocks may also have been influential, but here one may argue that that structure is perhaps not so unusual after all. Ljung and Selmer also looked briefly at planning problems in two other companies, and in both cases they noted subblocks of the same type—that is, consisting of upper and lower bounds on individual sales activities.

It also turns out that in the paperboard company, the rudiments of an iterative procedure for planning production and sales already exist (although that procedure does not conform to Dantzig–Wolfe decomposition). That is, the sales area heads submit tentative delivery plans to the headquarters group. These are then examined in the light of production possibilities and long-range marketing considerations, and a counterproposal for deliveries is sent to each sales area head. These proposals are revised once more by the sales area heads and then resubmitted to headquarters. Headquarters then decides on a final

delivery plan for each sales area. The breakdown of this plan to individual customers is carried out by the sales area head.

The conclusion is that the Dantzig–Wolfe decomposition method could be used as a planning tool in the paperboard company, since it seems to produce reasonable solutions and is somewhat reminiscent of iterative procedures already in use.* In any case, the application of a planning procedure founded on Dantzig–Wolfe decomposition in that company cannot be discarded out of hand as a naive theoretical idea.

6.3 A SIMULATION STUDY OF PLANNING PROCEDURES BASED ON THE DANTZIG–WOLFE AND TEN KATE METHODS IN A SLAUGHTERHOUSE

6.3.1 THE PLANNING PROBLEM OF THE SLAUGHTERHOUSE

The second case study is taken from the work of Christensen and Obel (1976) on a planning problem in a Danish slaughterhouse. The slaughterhouse slaughters pigs and produces various pork products. It is a cooperative corporation and has to accept all pigs delivered by member farmers. How best to utilize these pigs is thus a short-term decision problem—that is, what particular products to supply (e.g., ham, bacon, sausages) and in what quantities. The following planning problem formulation was obtained from the slaughterhouse, where it is used on a regular basis for short-run (one week) production planning. That is, the planning model is run regularly, in a single-level fashion (as an ordinary LP problem). Christensen and Obel, however, utilize that planning problem to simulate the application of two decomposition methods as a basis for planning procedures: the Dantzig–Wolfe and ten Kate methods.

The slaughterhouse is divided into functional departments for purchases, production, and sales. There are actually several production departments, but they may be considered as one for the purposes of this study. There are also several sales departments (seven altogether), each covering a particular product group (like "fresh pork products" or "sausages").

The planning problem has a block-angular LP structure, as displayed in Figure 6.3. The activities associated with the purchase, production, and sales departments are self-explanatory. The corporate constraints are all of the balance type, the balances referring to transfers from the purchase department to the production department, and transfers from the production to the sales departments. The departmental constraints of the purchase department all

* In the other two companies studied by Ljung and Selmer, iterative planning procedures reminiscent of Dantzig–Wolfe decomposition—or other decomposition methods—were apparently not in use.

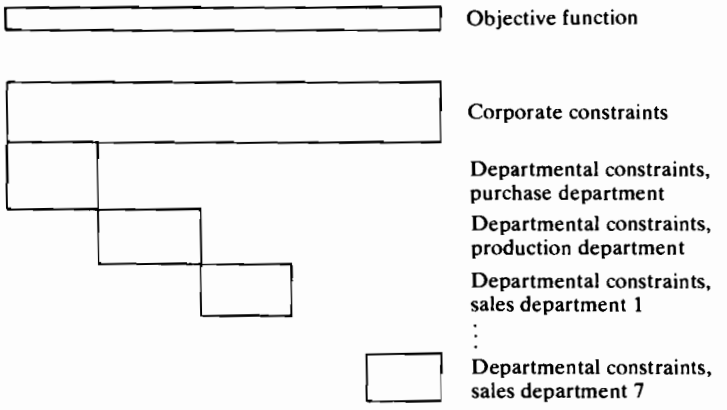


FIGURE 6.3 Coefficient matrix for the slaughterhouse planning problem.

consist of lower and upper bounds on individual purchase activities. These activities refer to the acquisition of live pigs (over and above the amounts that the slaughterhouse is forced to acquire from member farms) but also to parts of pigs that can be bought from other slaughterhouses to complement the acquisitions of live pigs. Certain other items, such as spices, are also bought. The local constraints of the sales departments also consist of only upper and lower bounds on individual sales activities. These bounds are mainly upper ones, expressing estimated sales possibilities. However, there are also some lower bounds, to account for existing delivery contracts. The local constraints of the production department consist of balance expressions for products produced and utilized exclusively within the production sphere. There are no constraints on physical production capacity, presumably implying that sufficient capacity is always available. The only limiting physical resources are the raw materials, meaning in particular the pigs, which can be obtained within certain bounds only (as mentioned earlier). Altogether, there are 606 variables and 575 constraints, of which 184 are corporate. The objective function is one of maximizing contribution to profit.

It may be noted here that upper and lower bounds on individual activities figure importantly among the local constraints, just as in Ljung and Selmer (1975).

For the purpose of applying some decomposition method as a planning tool, the above overall problem must be partitioned among the various organizational subunits. That is, one must decide which organizational subunits are to participate in the planning process and which part of the total problem each subunit is supposed to possess information about. One must, in effect, decide on an organizational structure to be used for the purpose of implementing the planning method. One such structure has already been indicated above,

namely to involve all nine departments (purchasing, production, and seven sales departments). This assumes that each department knows its own departmental constraints, and probably also its own objective function coefficients. In that case, headquarters will assume only the task of balancing deliveries between purchasing and production, and between production and sales. We will refer to this structure as No. 1.

A second possibility is to combine purchasing and production into one department but keep the sales departments. This implies that transfers from purchasing to production can be coordinated internally, within this purchasing/production department. The number of corporate constraints then diminishes from 184 to 149. We refer to this structure as No. 2.

A third possibility is to combine purchasing and production with the corporate constraints and to keep only the sales departments. This assumes that headquarters has complete information about the purchasing situation and about the production technology. More specifically, suppose the overall problem of the slaughterhouse can be written as (6.1) above, with $n = 8$. Let $j = 1 \ldots 7$ refer to sales activities in the seven sales departments, and $j = 8$ to purchasing and production activities. Then headquarters must know A_8, B_8, and c_8. Headquarters now has to plan both purchasing and production *and* balance the amounts produced with amounts to be sold by the sales areas. We refer to this structure as No. 3.

It can be seen that these three structures imply different organizational arrangements, and a different division of labor between organizational subunits in the planning process. It will be recalled that two different structures, also implying different organizational arrangements, were discussed in the case study of Ljung and Selmer, too. Here, however, the different structures arise as a consequence of *different two-level subproblem hierarchies relating to the same overall problem.* In the earlier case study, they arise as a consequence of *different overall problems*—that is, different formalizations of the underlying problem situation.

6.3.2 SIMULATED RESULTS USING THE DANTZIG–WOLFE METHOD AS A PLANNING PROCEDURE

The three organizational structures 1–3 discussed in the preceding subsection were all considered in the Dantzig–Wolfe simulation experiments. It turned out to be totally impossible to apply the Dantzig–Wolfe method as a planning procedure under the first two structures. * A variety of heuristic methods were tried for selecting a set of initial prices. It was hoped that some such set of prices would generate an initial set of department proposals resulting immediately

* It follows from the discussion in section 6.3.1 that under structure 1, there are 9 infimal subproblems, and the restricted master problem has 184 corporate constraints. Under structure 2, there are 8 infimal subproblems, and the restricted master problem has 149 corporate constraints.

(i.e., in the first iteration of the adjustment phase) in a feasible restricted master problem. In no case did this succeed. Therefore, a Phase I procedure was tried. This also had no positive effect. In no case was a feasible solution to the restricted master problem attained in less than 11 iterations.

A different starting strategy was then tried. The restricted master problem was supplied at the outset with various feasible starting solutions, composed from randomly generated departmental proposals. The objective function values of the initial restricted master problem were in all cases negative. After 20 iterations, that value increased typically by only 0.5 percent.

It is thus clear that it is impossible to use the Dantzig–Wolfe method as the basis for production and sales planning in this company, if organizational structures 1 or 2 are utilized for the planning process. These negative results may be explained as follows: The iterative information exchange between headquarters and departments in the adjustment phase of the Dantzig–Wolfe method supplies the restricted master problem with information about local conditions in the departments. In this case, the local conditions pertaining to the production department are somewhat complicated (i.e., many variables and constraints). It is simply not possible to supply the restricted master problem with sufficient information about the production department in a small number of iterations. It may be remarked that structures 1 and 2 gave results that were nearly equally bad, although the coordination problem facing headquarters is somewhat smaller in structure 2.

The Dantzig–Wolfe method worked somewhat better in conjunction with structure 3. Under that structure, purchasing and production activities are assigned to headquarters. This means that there are seven infimal subproblems (one for each sales department). The supremal subproblem in iteration t of the adjustment phase may be written as follows:

$$\begin{aligned}
&\text{Maximize} \quad \sum_{j=1}^{7} \left\{ \sum_{s=0}^{t-1} (c_j x_j^s)\lambda_j^s \right\} + c_8 x_8 \\
&\text{s.t.:} \quad \sum_{j=1}^{7} \left\{ \sum_{s=0}^{t-1} (A_j x_j^s)\lambda_j^s \right\} + A_8 x_8 \leq a,\ B_8 x_8 \leq b_8,\ x_8 \geq 0, \qquad (6.2) \\
&\qquad \sum_{s=0}^{t-1} \lambda_j^s = 1\ (j = 1 \ldots 7),\ \lambda_j^s \geq 0\ (\text{all } s \text{ and } j).
\end{aligned}$$

In (6.2), $j = 1 \ldots 7$ denotes sales departments, and x_8 is the vector of purchasing and production activities. x_j^s is the proposal obtained from sales department j in iteration s of the adjustment phase (it is assumed that the x_j^s correspond to extreme point, not extreme ray, solutions to the infimal subproblems). We note that under structure 3, the parts of the overall planning problem corresponding to purchasing and production activities have been put directly into the supremal subproblem. The sales departments only demand products from the

production department—i.e., they supply no products. This means that a simple starting strategy is available for generating an initial feasible restricted master problem: set the initial prices associated with the corporate constraints "high." This results in sales proposals from the sales departments where each individual sales variable is at its lower bound, if the initial prices are high enough. Unless the overall production and sales planning problem has no feasible solution at all, this must necessarily bring about a feasible solution to the restricted master problem in the first iteration.

Some different sets of "high" starting prices were tried. In all cases, a feasible solution to the restricted master problem was obtained immediately. However, the Dantzig–Wolfe method brought only slow improvement in the restricted master problem solution value. The initial iterations resulted in large negative solution values. After about six iterations, a small positive value had been attained, and after eleven iterations, about 90 percent of the true optimal solution value was attained. This is too slow if the Dantzig–Wolfe method is to be applied as a planning tool in a real company—as mentioned earlier, one would like to obtain a good result in no more than four iterations of information exchange.

6.3.3 SIMULATED RESULTS USING THE TEN KATE METHOD AS A PLANNING PROCEDURE

The economic interpretation of the ten Kate method (that is, the Benders method applied to block-angular LP problems) has been briefly discussed earlier (section 3.5.4). This interpretation indicates how the method could be used as the basis for a planning procedure. In iteration t of the adjustment phase, headquarters assigns a tentative allocation vector a_j^t of jointly utilized scarce resources and semifinished products to department j. Department j then solves the infimal subproblem

$$\begin{aligned} &\text{Maximize} && c_j x_j \\ &\text{s.t.:} && A_j x_j \leq a_j^t,\ B_j x_j \leq b_j,\ x_j \geq 0. \end{aligned} \tag{6.3}$$

If (6.3) has an optimal solution, then department j responds with the following information: the optimal solution value and the multiplier vector associated with the restrictions $A_j x_j \leq a_j^t$. This multiplier vector contains information about how the optimal solution value would change, if a_j^t were changed. With these two pieces of information, headquarters can construct a constraint of the type $z_j \leq (u_j^1, u_j^2)^p (a_j, b_j)$, to add to the supremal subproblem [see (3.37)]. If (6.3) is not feasible, then the information sent back to headquarters includes an extreme ray of the dual problem of (6.3).

It was shown in section 3.5.5 that the Dantzig–Wolfe and ten Kate methods may be regarded as dual methods. That is, if one takes the dual of the overall problem (6.1) and applies Dantzig–Wolfe decomposition to that dual, that is equivalent to applying ten Kate decomposition to the primal problem (6.1). When one is using Dantzig–Wolfe decomposition as a planning procedure, it is important to provide a good set of initial prices, enabling the construction of a feasible restricted master problem from the first set of proposals returned from departments. The duality analogue of this for the ten Kate method is that the initial tentative allocations sent to departments from headquarters must be such that the information returned from the infimal subproblems, when transformed into constraints for the supremal subproblem, produces a supremal subproblem with a bounded solution value.

One necessary condition for this is that the initial tentative allocations sent to departments from headquarters result in feasible infimal subproblems. If that is not the case, the supremal subproblem of the ten Kate method at the first iteration* will obviously have an unbounded solution value [since, for at least one index j, there are no restrictions of the type $z_j \leq (u_j^1, u_j^2)^p(a_j, b_j)$]. However, this is that the set of allocation vectors $a_1, a_2 \ldots a_n$ satisfying $\sum a_j = a$ is not sent to departments from headquarters does yield feasible infimal subproblems, there is nevertheless no guarantee that the resulting supremal subproblem in the first iteration has a bounded solution value. The reason for this is that the set of allocation vectors $a_1, a_2, \ldots, a_n$ satisfying $\sum a_j = a$ is not bounded. In fact, even if one starts out with *optimal* allocation vectors, there is no guarantee that the supremal subproblem in the first iteration has a bounded solution value.

For the slaughterhouse planning problem, using organizational structures 1 and 2, and a variety of initial allocations, including optimal ones, it was not possible to obtain a bounded optimal solution to the supremal subproblem in ten iterations. This is reminiscent of the Dantzig–Wolfe experiments with the same two structures, where it was not possible to obtain a feasible restricted master problem in fewer than 11 iterations. Again, the basic difficulty is that the supremal subproblem of the ten Kate method (as well as that of the Dantzig–Wolfe method) is attempting to collect information about the infimal subproblems. However, the production department subproblem is not trivial, and hence it is difficult to collect sufficient information in a small number of iterations.

Organizational structure 3 was also used in conjunction with the ten Kate method. As before, let the index $j = 1 \ldots 7$ refer to sales departments and $j = 8$ to the production and purchasing departments. In the notation of section 3.5.4,

* The iteration count is as follows: The initial set of tentative allocation vectors is announced in iteration 0. The supremal subproblem is then solved for the first time at the beginning of iteration 1.

the equivalent full master problem under the ten Kate method may be written as follows, if structure 3 is utilized [see (3.37)].

$$
\begin{aligned}
\text{Maximize} \quad & z_1 + \cdots + z_7 + c_8 x_8 \\
\text{s.t.:} \quad & a_1 + \cdots + a_7 + a_8 = a, \\
& A_8 x_8 - a_8 \leq 0,\ B_8 x_8 \leq b_8,\ x_8 \geq 0, \\
& z_1 \leq (u_1^1, u_1^2)^p (a_1, b_1) \quad (p = 1 \ldots P(1)), \\
& \vdots \\
& z_7 \leq (u_7^1, u_7^2)^p (a_7, b_7) \quad (p = 1 \ldots P(7)), \\
& (\tilde{u}_1^1, \tilde{u}_1^2)^r (a_1, b_1) \geq 0 \quad (r = 1 \ldots R(1)), \\
& \vdots \\
& (\tilde{u}_7^1, \tilde{u}_7^2)^r (a_7, b_7) \geq 0 \quad (r = 1 \ldots R(7)).
\end{aligned}
$$

It is seen from this formulation that the purchasing and production activities will be incorporated directly into the supremal subproblem, as called for by structure 3.

When the ten Kate planning method was applied to the slaughterhouse problem under structure 3, the result was again negative (no bounded supremal subproblem in a reasonable number of iterations). Structure 3 obviously guarantees that the purchasing and production activity levels are feasible from the point of view of the departmental constraints of those departments (since those departments are joined with headquarters, i.e., put into the supremal subproblem). Infeasibilities must therefore occur in the sales departments' infimal subproblems. A sales department infimal subproblem can be infeasible only if the product amounts allocated by headquarters are not sufficient to cover the lower bounds. For that reason, a variant of the ten Kate method was tried. Before the procedure was begun, the supremal subproblem was supplied with information about the sum of the lower bounds for each product and for each sales department. This information was incorporated as an additional set of restrictions in the supremal subproblem. Thus, one obtains a bounded supremal subproblem. It now turned out that feasible solutions to the overall planning problem with a total contribution of the order of 75 percent of the true optimal solution value could be obtained in four iterations.* This means that with this modification, a planning method based on the ten Kate method might be of interest for the slaughterhouse.

* Under structure 3, the resulting solution to the overall problem is recovered as follows, in the execution phase: Values for x_8 are given directly by the supremal subproblem. Values for $x_1 - x_7$ are obtained by means of the infimal subproblems (6.3) for $j = 1 \ldots 7$, with a_j^t given by the solution to the supremal subproblem in the last iteration of the adjustment phase.

6.3.4 SOME CONCLUSIONS

On the basis of the two cases, we may first conclude that the size of the underlying planning problem affects the applicability of a planning method founded on a decomposition algorithm. In particular, one would surmise that the number of corporate constraints is critical. The first study, by Ljung and Selmer, had only six corporate constraints. Of these, four were of a resource-allocation type, and two were product-mix constraints. In this situation, it was possible to find a good set of initial prices under the method based on Dantzig–Wolfe decomposition, namely "high" ones for the four resource constraints. Also, presumably because of the small number of corporate constraints, rapid improvement in the resulting solution value was obtained in only a few iterations. In the second study (by Christensen and Obel), the given planning problem involved a substantially larger number of corporate constraints, with a correspondingly larger coordination task facing headquarters. This is particularly true under organizational structures 1 and 2. Under structure 3, simple starting strategies resulting immediately in a feasible supremal subproblem under the Dantzig–Wolfe method and a bounded supremal subproblem under the ten Kate method could be devised. However, for the Dantzig–Wolfe method, improvement in the supremal subproblem solution value was so slow that it is doubtful whether that method would be of much use as a planning tool.

In the second place, it was seen that the ten Kate method produced substantially better results under structure 3 in the Christensen–Obel study, if the supremal subproblem was modified to incorporate certain *a priori* restrictions on the allocations. This means that if one has some *a priori* information about bounds on feasible allocations, then that information should be incorporated directly into the supremal subproblem. This has the effect of preventing extreme reallocations by the supremal subproblem and hence avoiding infeasible departmental subproblems (since extreme reallocations often result in at least one infeasible departmental subproblem). Considering the duality relationship between the ten Kate and Dantzig–Wolfe methods, this suggests that one may also introduce bounds on the prices in the restricted master problem under the Dantzig–Wolfe method. Such bounds—on allocations under ten Kate, and on prices under Dantzig–Wolfe—may bring about better plans within three or four iterations. In any case, they will prevent strong oscillations of the tentative indices sent to departments or divisions (prices under Dantzig–Wolfe, and quantities under ten Kate) in the different iterations of information exchange during the adjustment phase. Both decomposition methods may otherwise give rise to strong oscillations, which may have a confusing effect on the organizational subunits participating in the planning process.

The upshot of this last consideration is that the more *a priori* information (e.g., learning effects from earlier periods) about the total planning problem

headquarters has, the better the planning methods will work. More precisely, the information at hand *a priori* can be used to construct suitable bounds on the allocations under ten Kate and on the prices under Dantzig–Wolfe.

6.4 FINAL REMARKS ON PLANNING PROCEDURES BASED ON DECOMPOSITION METHODS

The two case studies presented in this chapter may appear somewhat strange. A practically inclined reader might well argue that planning in real-world corporations will never be carried out in the fashion simulated in these two case studies. Yet in the literature the opposite is suggested. As stated in section 6.1.2, there is a very large literature dealing with planning methods founded on decomposition schemes. If this literature is to be taken seriously, it is necessary to move in the direction of applying these theoretical planning methods to concrete planning problems. Perhaps for good reasons, apparently no real-world company has dared to install such a planning method. However, if one cannot actually implement such planning methods as a research experiment, one can at least simulate their behavior in the context of realistic planning problems. That is what these two studies do, and that is their significance.

We will encounter yet another simulation study of a similar kind later in this volume, in Chapter 10 on water pollution control. It has been suggested that planning procedures based on decomposition methods could be used for decision making as regards pollution control. From a formal point of view, an overall problem of planning effluent levels for a set of polluters so as to minimize some total cost function is actually rather similar to the overall problem (6.1) of this chapter. This means that planning methods of the type discussed here could in principle be applied in the pollution control planning situation, too.

Finally, two current research directions related to the discussion in this chapter will be pointed out. In the first place, the discussion throughout has assumed that departments are participating honestly and unselfishly in the planning procedures, meaning in particular that they send unbiased and "true" information to headquarters in each iteration of the adjustment phase. However, suppose that departments (or divisions) pursue some private objectives of their own, and that they send false, or biased, messages to headquarters. What effect will this have on the outcome of the planning process? This situation, where false or biased information is sent during the adjustment phase, has been referred to as divisional (or departmental) cheating. It is discussed by Jennergren and Müller (1973), among others.

In the second place, the slaughterhouse case study discussed the use of planning methods founded on decomposition methods under different organizational structures. That is, given that one has decided on a general type of planning procedure (e.g., based on Dantzig–Wolfe decomposition), there

remain certain further decisions, such as: How many and which departments (or divisions) are to participate in the planning process? These further decisions define an organizational structure. Defining an organizational structure for the purpose of using a formalized planning method of the kind treated in this chapter certainly is not equivalent to the total task of organizational design. Nevertheless, attempts have been made to derive principles for good organizational design in general from studies of the type mentioned here (Baligh and Burton 1976; Obel 1978).

REFERENCES

Almon, C. 1963. Central planning without complete information at the center, pp. 462–466. In G. B. Dantzig, Linear Programming and Extensions. Princeton, New Jersey: Princeton University Press.

Atkins, D. 1974. Managerial decentralization and decomposition in mathematical programming. Operational Research Quarterly 25: 615–624.

Bailey, F. N. 1976. Decision processes in organizations, pp. 82–111. In R. Saeks (ed.), Large-Scale Dynamical Systems. North Hollywood, California: Point Lobos Press.

Baligh, H. H., and R. M. Burton. 1976. Organization structure and cooperative market relations. Omega 4: 583–593.

Baumol, W. J., and T. Fabian. 1964. Decomposition, pricing for decentralization, and external economies. Management Science 11: 1–32.

Burton, R. M., W. W. Damon, and D. W. Loughridge. 1974. The economics of decomposition: Resource allocation vs transfer pricing. Decision Sciences 5: 297–310.

Burton, R. M., and B. Obel. 1977. The multilevel approach to organizational issues of the firm: A critical review. Omega 5: 395–414.

Charnes, A. R., R. W. Clower, and K. O. Kortanek. 1967. Effective control through coherent decentralization with preemptive goals. Econometrica 35: 294–320.

Christensen, J., and B. Obel. 1976. Simulation of Decentralized Planning in Two Danish Organizations Using the Decomposition Scheme from Linear Programming. Social Science Report Series No. 37. Odense, Denmark: Odense University.

Ennuste, Iu. A. 1972. Problems of decomposition analysis of optimal planning tasks. (In Russian.) Ekonomika i matematicheskie metody 8: 535–545.

Freeland, J. R. 1973. Conceptual Models of the Resource-Allocation Process in Hierarchical Decentralized Organizations. Ph.D. dissertation. Georgia Institute of Technology.

Freeland, J. R., and N. R. Baker. 1975. Goal partitioning in a hierarchical organization. Omega 3: 673–688.

Hass, J. E. 1968. Transfer pricing in a decentralized firm. Management Science 14: B-310–B-331.

Jennergren, L. P. 1971a. Studies in the Mathematical Theory of Decentralized Resource Allocation. Ph.D. dissertation. Stanford University.

Jennergren, L. P. 1971b. Mathematical programming models of decentralized budgeting procedures. Swedish Journal of Economics 73: 417–426.

Jennergren, L. P. 1972. Decentralization on the basis of price schedules in linear decomposable resource-allocation problems. Journal of Financial and Quantitative Analysis 7: 1407–1417.

Jennergren, L. P. 1973. A price schedules decomposition algorithm for linear programming problems. Econometrica 41: 965–980.

Jennergren, L. P. 1975. Decentralization in Organizations. Social Science Report Series No. 14. Odense, Denmark: Odense University [To be published in P. Nystrom and W. H. Starbuck (ed.), Handbook of Organizational Design. London: Oxford University Press.]

Jennergren, L. P., and W. Müller. 1973. Simulation experiments of resource-allocation decisions in two-level organizations. Social Science Research 2: 333–352.

ten Kate, A. 1972. Decomposition of linear programs by direct distribution. Econometrica 40: 883–898.

Kornai, J. 1975. Mathematical Planning of Structural Decisions. 2nd ed. Amsterdam: North-Holland.

Kornai, J., and T. Kiptak. 1965. Two-level planning. Econometrica 33: 141–169.

Kulikowski, R. 1975. Decentralized management and optimization of development in large production organizations. Control and Cybernetics 4: 5–18.

Kydland, F. 1975. Hierarchical decomposition in linear economic models. Management Science 21: 1029–1039.

Ljung, B., and J. Selmer. 1975. Samordnad planering i decentraliserade företag. (Coordinated Planning in Decentralized Corporations, in Swedish.) Stockholm: Bonniers.

Malinvaud, E. 1967. Decentralized procedures for planning, pp. 170–208. In E. Malinvaud and M. O. L. Bacharach (ed.), Activity Analysis in the Theory of Growth and Planning. London: Macmillan.

Mandel', A. B. 1973. Internal prices in the control of industrial firms. (In Russian.) Ekonomika i matematicheskie metody 9: 500–513.

Martines-Soler, F. 1974. Mechanisms for guiding processes of working out the plan in systems of optimal planning (In Russian), pp. 154–171. In Matematicheskie metody v ekonomicheskikh issledovaniiakh. Moscow: Nauka.

Obel, B. 1978. On organizational design—From a linear programming point of view. Journal of Management Studies 15: 123–137.

Pervozvanskaia, T. N., and A. A. Pervozvanskii. 1966. The distribution of centralized resources among many firms. (In Russian.) Ekonomika i matematicheskie metody 2: 682–689.

Polterovich, V. M. 1969. Block methods of concave programming and their economic interpretation. (In Russian.) Ekonomika i matematicheskie metody 5: 865–876.

Polterovich, V. M. 1972. Principles for the optimization of block structures (In Russian), pp. 443–448. In N. P. Fedorenko (ed.), Problemy optimal'nogo funktionirovaniia sotsialisticheskoi ekonomiki. Moscow: Nauka.

Ruefli, T. W. 1974. Analytic models of resource allocation in hierarchical multi-level systems. Socio-Economic Planning Sciences 8: 353–363.

Zschau, E. V. W. 1967. A primal decomposition algorithm for linear programming. Ph.D. dissertation. Stanford University.

7 Operations Management

7.1 INTRODUCTION AND OVERVIEW

The area of operations management is characterized by complex decision-making processes. Establishing production levels for many different items so as to meet demand over a given planning horizon while keeping the inventories at acceptable levels is a decision problem that can be solved only by making various simplifying assumptions about cost structure, demand patterns, and other aspects. This inherent complexity offers an interesting challenge for multilevel techniques.

One of the major concerns in the area of operations management has been to formulate aggregate production planning models, that is, models for the simultaneous determination of production, inventories, and work force (regular and overtime). The most celebrated aggregate planning model is that of Holt *et al.* (1960) (based on quadratic costs and thus yielding linear decision rules as functions of the demand forecasts). Among other aggregate models are those proposed by Jones (1967) (parametric production planning), and by Taubert (1968) (search decision rules). An aggregate production planning model could, in our terminology, correspond to the supremal subproblem. The infimal subproblems would then correspond to the associated disaggregated problems. Disaggregation, however, has received little attention, although the model of Holt *et al.* is an exception. The lack of a comprehensive treatment of the various disaggregation issues reveals the absence of a widespread multilevel approach in the area of operations management.

To simplify the exposition in this chapter, we will not consider work force planning, although manpower aspects could have been included without major difficulties. Instead, we will consider two alternative multilevel approaches to a standard multiproduct lot-size scheduling problem (i.e., a problem of planning

production and inventories) which may be written as:

$$\begin{aligned}
&\text{Minimize} && \sum_{i=1}^{n}\sum_{t=1}^{T}(C_{it}(x_{it})+h_{it}I_{it}) && \\
&\text{s.t.:} && I_{i,t-1}+x_{it}-I_{it}=r_{it} && (i=1\ldots n,\ t=1\ldots T), \\
& && I_{i0}=0 && (i=1\ldots n), \qquad (7.1)\\
& && \sum_{i=1}^{n} l_i x_{it}\le L_t && (t=1\ldots T), \\
& && x_{it},\ I_{it}\ge 0 && (i=1\ldots n,\ t=1\ldots T).
\end{aligned}$$

Problem (7.1) is the overall problem of this chapter. It has a straightforward interpretation. Let x_{it} denote the amount produced of product i in period t and I_{it} the number of units of inventory left of product i at the end of period t $(i=1\ldots n,\ t=1\ldots T)$. For a given requirements schedule $\{r_{it}\}$, the first set of constraints are then simply the inventory balance equations for some given initial inventory levels. The production of one unit of product i requires l_i units of a common resource (say, some raw material), of which L_t units are available in period t.* This explains the resource constraints. The nonnegativity restriction on the inventory levels rules out back orders. The first component of the objective function, the production cost, is of the following form:

$$C_{it}(x_{it})=\begin{cases}0 & \text{if } x_{it}=0\\ s_{it}+c_{it}x_{it} & \text{if } x_{it}>0 \quad (s_{it}>0).\end{cases} \qquad (7.2)$$

The linear part of the objective function accounts for the inventory costs.

Problem (7.1) is one of the simplest of the many models that have been proposed in the area of production scheduling. It is, for instance, possible to include more than one type of resource constraint, to introduce labor costs (e.g., regular time and overtime) and corresponding manpower decision variables. These extensions would not alter the following developments in a significant way.

Throughout we assume the existence of a solution to (7.1). It is noted that we also assume that $I_{i0}=0$. This may be interpreted to mean that delivery requirements have been "netted" by deducting initial inventories.

Problem (7.1) may be a difficult nonlinear programming problem, especially if n is large. In subsequent sections, we outline two multilevel methods for solving (7.1). Both methods result merely in "good" (i.e., nonoptimal) solutions. In section 7.2 we will present a column generation approach that was

* It is most natural to imagine that the resource constraints do not refer to labor availabilities. The reason is that a labor availability constraint would typically involve setup times as well as variable production times. That is, a labor availability constraint would not be linear, but nonlinear [like the production cost (7.2)].

suggested by Dzielinski and Gomory (1965) based on an approximation of the original problem proposed by Manne (1958). Section 7.3 contains a different approach known as hierarchical production planning (designed by Hax, Meal and others at MIT) (Hax and Meal, 1975). We immediately point out that this latter approach is not multilevel in the sense of this volume, since an iterative interaction between subproblems on different levels is lacking. Nonetheless, it is multilevel in a more general sense. We have included hierarchical production planning here, since it offers an interesting alternative to the column generation approach. Moreover, there are actually some possibilities of incorporating interactive features in hierarchical production planning.

7.2 A COLUMN GENERATION APPROACH

7.2.1 AN APPROXIMATE LP PROBLEM

The underlying idea of the column generation approach is first to formulate a linear program approximating problem (7.1), as initially developed by Manne (1958) for a planning problem closely related to (7.1). One then applies column generation to the resulting LP problem.

Without loss of optimality, we may restrict ourselves to solutions to (7.1) such that $I_{iT} = 0$ for all i.* The following observation provides the key to the approximating linear program: The set of extreme points to a set of constraints

$$
\begin{aligned}
& I_0 = 0, \\
& I_{t-1} + x_t - I_t = r_t \qquad (t = 1 \ldots T), \\
& I_T = 0, \\
& x_t, I_t \geq 0 \qquad\qquad (t = 1 \ldots T)
\end{aligned}
$$

is precisely the set of *dominant schedules*—i.e., all those solutions such that for all t, $I_{t-1}x_t = 0$. Dominant schedules are often referred to as Wagner–Whitin schedules after Wagner and Whitin (1958).

Suppose that, for each given product i, the set of dominant schedules is given by $\{(x_{i1}^j \ldots x_{iT}^j, I_{i1}^j \ldots I_{iT}^j);\ j = 1 \ldots J(i)\}$. If we now introduce the notation

$$
d_i^j = \sum_{t=1}^{T} (C_{it}(x_{it}^j) + h_{it} I_{it}^j),
$$

and

$$
L_{it}^j = l_i x_{it}^j,
$$

* This restriction involves optimality loss only if some production and/or holding costs are strictly negative.

and if we restrict the solution set of (7.1) to those feasible solutions that can be expressed as convex combinations of the dominant schedules of the individual products, we obtain the following linear program:

$$\begin{aligned} \text{Minimize} \quad & \sum_{i=1}^{n} \sum_{j=1}^{J(i)} d_i^j \theta_i^j \\ \text{s.t.:} \quad & \sum_{i=1}^{n} \sum_{j=1}^{J(i)} L_{it}^j \theta_i^j \leq L_t \qquad (t = 1 \ldots T), \\ & \sum_{j=1}^{J(i)} \theta_i^j = 1 \qquad (i = 1 \ldots n), \\ & \text{all } \theta_i^j \geq 0. \end{aligned} \tag{7.3}$$

Every feasible solution to (7.3) defines a feasible solution to (7.1). If an integer-valued solution is optimal for (7.3), then the corresponding solution to (7.1) is also optimal.* If $n > T$, an optimal basic solution to (7.3) contains at least $n - T$ integer-valued basic variables (this can be seen by a simple counting argument). If we make the plausible assumption that n is much larger than T, the optimal solution to (7.3) will be "almost integer" and hence, heuristically, the corresponding solution to (7.1) will be "almost optimal" for (7.1).

The number of dominant schedules for each product can be quite large. An upper bound is given by 2^{T-1}. This fact, plus the fact that the number of products is also often quite large, makes it desirable to use a column generation technique, rather than to generate all dominant schedules in advance.

7.2.2 GENERATION OF DOMINANT SCHEDULES AND A TWO-LEVEL ALGORITHM

It will now be demonstrated how dominant schedules, or columns, can be generated for problem (7.3). Let the simplex multiplier vector pertaining to the resource restrictions associated with some basis be $\pi = (\pi_1 \ldots \pi_t \ldots \pi_T)$. It is easy to see that $\pi \leq 0$. Let α_i $(i = 1 \ldots n)$ be the simplex multiplier associated with the ith convexity constraint. To identify a possible new dominant schedule for product i, one solves the following optimization problem.

$$\text{Minimize} \quad \left(d_i^j - \sum_{t=1}^{T} \pi_t L_{it}^j - \alpha_i \right) \quad \text{over } j = 1 \ldots J(i),$$

or, since α_i is independent of j,

$$\text{Minimize} \quad \left(d_i^j - \sum_{t=1}^{T} \pi_t L_{it}^j \right) \quad \text{over } j = 1 \ldots J(i). \tag{7.4}$$

* It is not correct, as is sometimes suggested in the literature, that if all θ_i^j are restricted to be integers in (7.3), an optimal solution to this integer programming problem always defines an optimal solution to (7.1).

This problem can be solved efficiently by a forward dynamic programming algorithm.*

The optimum in (7.4) is found (temporarily dropping the index i) by considering the problem

$$\begin{aligned} \text{Minimize} \quad & \sum_{t=1}^{T} (v_t x_t + s_t \delta(x_t) + h_t I_t) \\ \text{s.t.:} \quad & I_{t-1} + x_t - I_t = r_t \quad (t = 1 \ldots T), \\ & I_0 = 0, I_T = 0, \\ & \text{all } x_t, I_t \geq 0, \end{aligned} \tag{7.5}$$

where $v_t = c_t - \pi_t l$, $\delta(x_t) = 0$ if $x_t = 0$ and 1 otherwise [problem (7.5) has a concave objective function; hence it has an optimum at an extreme point, a dominant schedule]. The reader familiar with elementary inventory theory will recognize problem (7.5) as the single-product dynamic lot-size problem introduced by Wagner and Whitin (1958). The following dynamic programming recursion will find a dominant schedule optimizing (7.5) and hence solving (7.4). It is based on the observation that, since only dominant schedules have to be considered, a positive production quantity in a particular period t corresponds to the cumulative requirements of q successive periods starting from period t with $1 \leq q \leq T - t + 1$. Let $f(t)$ be cost associated with an optimal dominant schedule for the periods 1 to t. Then, since $I_0 = 0$,

$$f(1) = s_1 + v_1 r_1,$$

and for $2 \leq t \leq T$

$$f(t) = \text{minimum} \begin{cases} s_t + v_t r_t + f(t-1) \\ \min\limits_{1 \leq u < t} \left\{ s_u + v_u R(u, t) + \sum\limits_{k=u}^{t-1} h_k R(k+1, t) + f(u-1) \right\} \end{cases} \tag{7.6}$$

where $R(x, y) = \sum_{t=x}^{y} r_t$. Once $f(T)$ has been computed, an optimal dominant schedule can be retrieved by an obvious backtracking scheme.

A two-level algorithm can now be described. Suppose that, at some iteration of the adjustment phase, subsets $\mathcal{J}(i)$ of dominant schedules have been generated. The supremal subproblem consists in solving (7.3) with the summations over all dominant schedules replaced by the index sets $\mathcal{J}(i)$. The supremal subproblem reports the dual multipliers α_i and π to each infimal subproblem i. Each infimal subproblem is solved by a dynamic programming recursion of the type (7.6). If $d_i^j - \sum \pi_t L_{it}^j - \alpha_i < 0$ for the identified schedule, it is reported back to the supremal subproblem, where it is added as a new column.

* The nontechnically oriented reader can skip the explanation of the dynamic programming algorithm and continue with the description of the two-level method.

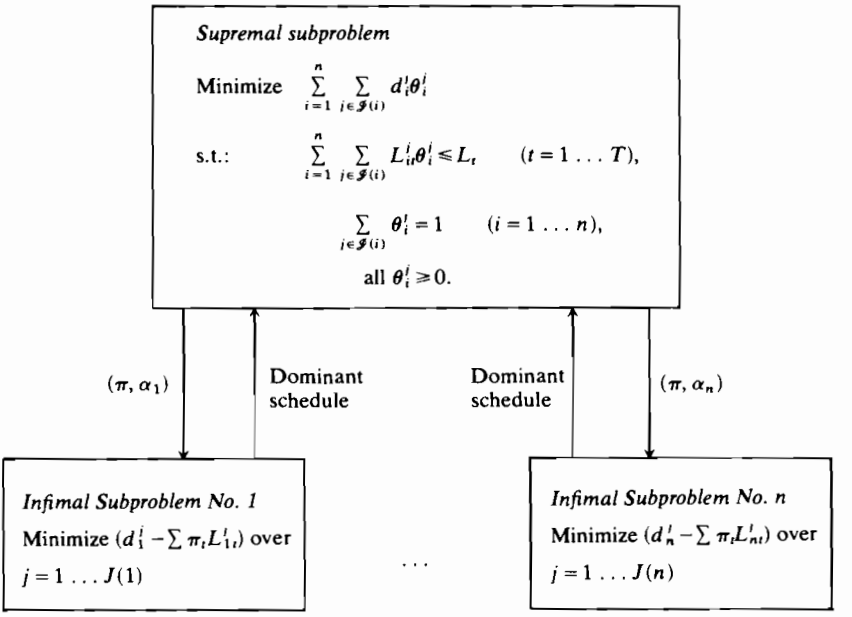


FIGURE 7.1 The adjustment phase of the column generation algorithm for lot-size production scheduling.

Such a column has $T+n$ elements with its first T elements given by L_{it}^j. The remaining n entries are zero, except for unity in the ith position. The associated objective function coefficient is given by d_i^j. If, at some iteration, no infimal subproblem can identify a dominant production schedule such that $d_i^j - \sum \pi_t L_{it}^j - \alpha_i < 0$, no new column gets added to the supremal subproblem. This means that an optimal solution to problem (7.3) has been found, and so the adjustment phase terminates. In the execution phase a solution to the original problem (7.1) is recovered from the supremal subproblem. As already pointed out, that solution to (7.1) need not be an optimal one. This means that the two-level subproblem hierarchy is not equivalent to the original problem (7.1). That is, coordinability does not hold (see section 2.2.1). The interaction between the subproblems in the adjustment phase is displayed in Figure 7.1.

7.2.3 APPLICATIONS

Dzielinski and Gomory (1965) applied column generation to solve some experimental test problems of lot-size production scheduling. Actually, their problems were more complex than the one described here: for instance, they allowed several classes of labor. They reported encouraging computational results and stated that some of the test problems were so large that they could

not have been handled by ordinary linear programming, necessitating a two-level method (Dzielinski and Gomory 1965, p. 888).

Lasdon and Terjung (1971) considered a problem related to (7.1) but again more complex in nature. They reported successful implementation at various plants of a major U.S. tire manufacturer. Their algorithm uses column generation as well as a generalized upper bounding procedure as applied to the constraints of the form $\sum_{j=1}^{J(i)} \theta_i^j = 1$, $i = 1 \ldots n$.

However, other authors have solved lot-size production scheduling problems by ordinary linear programming. Gorenstein (1970), for example, considered a tire production scheduling problem. He notes that while the problem could have been solved by a two-level method (p. B-75), he actually did use a single-level method, direct linear programming.

7.3 HIERARCHICAL PRODUCTION PLANNING

7.3.1 INTRODUCTION TO HIERARCHICAL PRODUCTION PLANNING

Formally, hierarchical production planning is a heuristic solution procedure for solving optimization problems of type (7.1). However, this hierarchical approach was designed for problems not directly amenable to the mathematical programming techniques described in section 7.2. Hax and Meal (1975) report an implementation in which the number of products is of the order of 10,000, making an approach as described in the previous section impractical.

From the outset, we stress that a comprehensive treatment of what is known as the hierarchical approach to production planning as developed at MIT is beyond the scope of the present text, but would be appropriate in specialized texts on operations management. For the sake of completeness, however, we list some relevant research documents: Armstrong and Hax (1974); Hax and Meal (1975); Golovin (1975); Gabbay (1975); and Bitran and Hax (1976).

Since hierarchical production planning is more of an approach than a precise algorithm, a discussion of it can be made more concrete by considering a specific overall problem formulation. For that reason we have chosen to illustrate the method for the overall problem (7.1), while noting that more complex problem formulations present no great difficulties.

7.3.2 A THREE-LEVEL DISAGGREGATION SCHEME

What makes many manufacturing systems with batch-type production complex is the presence of a large number of products. However, as Hax and Meal point out, the product structure often allows for a useful hierarchical approach. Hax and Meal propose a three-level approach to the overall problem, identifying the following levels: *the item level*, the *item–family level* and the *product-type*

level. At the lowest level, the item level, all the final products are considered. At the intermediate level, the item–family level, those items are grouped together that require the same tooling and machine setups. Finally, at the highest level, the different item families are grouped into product types, so that all the item families contained in one product type have their production quantities determined by one aggregate production plan.

In order to relate the Hax–Meal disaggregation scheme to problem (7.1), let $N = \{1 \ldots n\}$ denote the set of all products. The product types are defined as those groups of products having identical variable production and holding costs (c_{it} and h_{it}), the same resource usage coefficients (l_i), and the same seasonal demand patterns (the importance of this will become clear later). If m product types result, we obtain m subsets of N, written as $N_1^1 \ldots N_j^1 \ldots N_m^1$, and constituting a partition of N. Products belonging to the same product type, say N_j^1, that have identical setup costs make up an item family. If product type j has p_j item families, we obtain p_j subsets of N_j^1, written as $N_1^{2j} \ldots N_k^{2j} \ldots N_{p_j}^{2j}$, forming a partition of N_j^1. At the third level, the item level, one trivially considers the n individual products. The scheme is depicted in Figure 7.2.

Whether such a disaggregation scheme is useful depends on the application at hand. In the application discussed by Hax and Meal, the 10,000 products could be split up into five product types and about 200 item families.

7.3.3 THE PRODUCT-TYPE-LEVEL SUBPROBLEM

One can formulate subproblems corresponding to each level identified above. Here we discuss the approach proposed by Bitran and Hax (1976). A different solution procedure is outlined by Hax and Meal (1975).

Making reference to the overall problem (7.1) we denote

$$c_{jt}^1 = c_{it},\ h_{jt}^1 = h_{it} \text{ and } l_j^1 = l_i \text{ for } i \in N_j^1,\ j = 1 \ldots m,\ t = 1 \ldots T.$$

Furthermore, let r_{jt}^1 be an aggregate demand forecast for product type j. We note that the quantities r_{jt}^1 need not be derived from the detailed estimates for the individual items. Of course, if such estimates are available, then $r_{jt}^1 = \sum_{i \in N_j^1} r_{it}$.

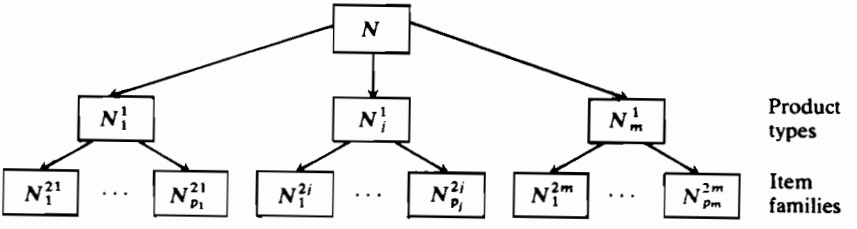


FIGURE 7.2 The Hax–Meal disaggregation scheme.

Bitran and Hax (1976) propose the following LP model as the product-type-level subproblem:

$$
\begin{aligned}
\text{Minimize} \quad & \sum_{j=1}^{m} \sum_{t=1}^{T} (c_{jt}^{1} x_{jt}^{1} + h_{jt}^{1} I_{jt}^{1}) \\
\text{s.t.:} \quad & I_{j,t-1}^{1} + x_{jt}^{1} - I_{jt}^{1} = r_{jt}^{1} && (j = 1 \ldots m,\ t = 1 \ldots T), \\
& I_{j0}^{1} = 0 && (j = 1 \ldots m), \qquad (7.7) \\
& \sum_{j=1}^{m} l_{j}^{1} x_{jt}^{1} \leq L_{t} && (t = 1 \ldots T), \\
& x_{jt}^{1},\ I_{jt}^{1} \geq 0 && (j = 1 \ldots m,\ t = 1 \ldots T).
\end{aligned}
$$

Problem (7.7) will usually be a much smaller problem than (7.1). At this level one is interested only in aggregate planning. Operational questions (e.g., scheduling) are relegated to the lower levels, explaining why setup costs have been deleted from the formulation.

The underlying idea of the present approach is that problem (7.7) is solved to determine an aggregate plan for the full planning horizon, whereas in the lower level subproblems operational details are considered whose complexity is reduced by restricting the relevant planning horizon to the present planning period (that is, $t = 1$). The actual system will operate over time by employing a rolling horizon at the first level. A critical and interesting issue is to determine a satisfactory length of the planning horizon. Indeed, if good solutions can be guaranteed by solving (7.7) for small T, considerable gains can be realized, not only in a computational sense but also in term of forecasting accuracy. Gabbay (1975) offers a discussion of these issues.

The first-level subproblem (7.7) is a straightforward LP model that, in itself, can be a useful tool for strategic planning. It will provide input data to the second level, the item-family level.

7.3.4 THE ITEM-FAMILY-LEVEL SUBPROBLEMS

Since we assume the existence of m product types, we will have m item-family-level subproblems, to each of which the product-type-level problem (7.7) will communicate the optimal production quantity of the first period (the present planning period), denoted as $\tilde{x}_{j1}^{1}$, $j = 1 \ldots m$. At this level, operational issues become dominant. For each item family k of a given product type j one assumes given a safety stock $\underline{I}_{k}^{2}$, $k = 1 \ldots p_{j}$,* to absorb inaccuracies in the demand forecasts and an overstock limit $\bar{I}_{k}^{2}$ to account for the maximum "reasonable" demand that can occur during the rest of the planning horizon (standard inventory-theoretic techniques can be used to determine these

* It will be implicitly assumed that we are dealing with a given product type j.

quantities). Furthermore, for each family the actual stocklevel, $\hat{I}_k^2$, at the beginning of the period is checked. Of course, unforeseen fluctuations in demand may make the values $\hat{I}_k^2$ quite arbitrary (e.g., $\hat{I}_k^2 < \underline{I}_k^2$). Finally, a demand forecast r_k^2 is assumed to be available. It should be pointed out that it is *not* assumed here that $\hat{I}_k^2 = 0$. This means that the demand forecast r_k^2 has *not* been "netted" by deducting initial inventory.

In the formulation of the item-family-level subproblem for product type j we will make use of the following classification of families within that product type:

$$K_j = \{k | \hat{I}_k^2 < r_k^2 + \underline{I}_k^2\}$$

and

$$K_j' = \{k | \hat{I}_k^2 \geq r_k^2 + \underline{I}_k^2\},$$

that is, a classification determining which families trigger (need to be produced) during the coming planning period. Production for families in K_j' will occur only after satisfaction of the requirements for the families in K_j within the overall constraint given by $\tilde{x}_{j1}^1$. It is clear that a lower and an upper bound on the production quantity of family k, x_k^2 are given by

$$\underline{x}_k^2 \equiv \max\{0, r_k^2 + \underline{I}_k^2 - \hat{I}_k^2\}$$

and

$$\bar{x}_k^2 \equiv \max\{0, r_k^2 + \bar{I}_k^2 - \hat{I}_k^2\},^*$$

respectively. Depending on the magnitudes of the quantities $\underline{x}_k^2$ and $\bar{x}_k^2$ *vis á vis* $\tilde{x}_{j1}^1$, the item-family-level subproblem specializes into three cases. We will denote the proposed solution by $\tilde{x}_k^2$, $k = 1 \ldots p_j$. Note again that that solution refers to the present planning period, i.e., $t = 1$.

Case 1. $\sum_{k \in K_j} \bar{x}_k^2 \leq \tilde{x}_{j1}^1$.

Then we set $\tilde{x}_k^2 = \bar{x}_k^2$ for $k \in K_j$ and the remaining capacity, if any, can be filled by sequentially setting production quantities of families in K_j' equal to their upper bound in increasing order of their run-out time.

Case 2. $\sum_{k \in K_j} \underline{x}_k^2 \geq \tilde{x}_{j1}^1$.

This is the case in which safety stocks will be violated and back orders may arise. Bitran and Hax suggest the following proportional allocation of capacity:

$$\tilde{x}_k^2 = \frac{\underline{x}_k^2}{\sum_{k \in K_j} \underline{x}_k^2} \tilde{x}_{j1}^1, \; k \in K_j.$$

* In principle $\hat{I}_k^2 > r_k^2 + \bar{I}_k^2$ is possible.

Case 3. $\sum_{k\in K_i} \underline{x}_k^2 < \tilde{x}_{i1}^1 < \sum_{k\in K_i} \bar{x}_k^2$.

This is the nontrivial case, and Bitran and Hax propose to solve the following optimization problem:

$$\begin{aligned} \text{Minimize} \quad & \sum_{k\in K_i} \frac{s_k^2 r_k^2}{x_k^2} \\ \text{s.t.:} \quad & \sum_{k\in K_i} x_k^2 = \tilde{x}_{i1}^1, \\ & \underline{x}_k^2 \le x_k^2 \le \bar{x}_k^2 \quad (k \in K_i), \end{aligned} \tag{7.8}$$

where $s_k^2 = s_{i1}$ for $i \in N_k^{2i}$ [see (7.2)].

The rationale of (7.8) is the following: all assigned production from the first level ($\tilde{x}_{i1}^1$) is allocated among the families that trigger (i.e., $k \in K_i$). The production runs will be highest for families with high setup costs and high demand forecasts. Since all families are supposed to have the same seasonal demand patterns, setting high production quantities for families with high demand forecasts takes into account demands in later planning periods as well. An algorithm that solves (7.8) is presented below. For the reader not interested in algorithmic developments, it is sufficient to observe that (7.8) is a convex programming problem of the knapsack type for which an efficient, finite algorithm exists. One may hence jump to section 7.3.5 without loss of continuity.

Consider now the algorithm for solving (7.8). For notational convenience we set $b_k = s_k^2 r_k^2$. The algorithm of Bitran and Hax is based on a relaxation of (7.8), where the constraints $\underline{x}_k^2 \le x_k^2 \le \bar{x}_k^2$ are deleted. Also, the right-hand side will be parameterized, and subsets of K_i will be considered. At some iteration v we have for some right-hand-side value y^v and some subset K^v of K_i the following optimization problem:

$$\begin{aligned} \text{Minimize} \quad & \sum_{k\in K^v} \frac{b_k}{y_k} \\ \text{s.t.:} \quad & \sum_{k\in K^v} y_k \le y^v, \\ & y_k \ge 0. \end{aligned} \tag{7.9}$$

An optimal solution to (7.9) is found by a straightforward application of Lagrangean techniques, and is given by:

$$y_k^v = \frac{\sqrt{b_k}}{\sum_{k\in K^v} \sqrt{b_k}} y^v, \; k \in K^v.$$

The algorithm can now be described:

Step 0. Let $v = 0$, $y^v = \bar{x}^1_{1j}$, $K^v = K_j$.

Step 1. Solve (7.9), to obtain $\{y^v_k\}$. If $\underline{x}^2_k \leq y^v_k \leq \bar{x}^2_k$ for $k \in K^v$, store the values $\{y^v_k\}$, and go to Step 4. Otherwise, go to Step 2.

Step 2. Determine

$$K^v_+ = \{k | k \in K^v,\ y^v_k \geq \bar{x}^2_k\},$$

and

$$K^v_- = \{k | k \in K^v,\ y^v_k \leq \underline{x}^2_k\}.$$

Compute

$$\Delta^+ = \sum_{k \in K^v_+} (y^v_k - \bar{x}^2_k)$$

and

$$\Delta^- = \sum_{k \in K^v_-} (\underline{x}^2_k - y^v_k).$$

Go to Step 3.

Step 3. If $\Delta^+ \geq \Delta^-$, reset $y^v_k = \bar{x}^2_k$ for $k \in K^v_+$. If $\Delta^+ < \Delta^-$, reset $y^v_k = \underline{x}^2_k$ for $k \in K^v_-$. Let

$$K^{v+1} = \begin{cases} K^v \backslash K^v_+ & \text{if } \Delta^+ \geq \Delta^-, \\ K^v \backslash K^v_- & \text{if } \Delta^+ < \Delta^-, \end{cases}$$

and

$$y^{v+1} = \begin{cases} y^v - \sum_{k \in K^v_+} y^v_k & \text{if } \Delta^+ \geq \Delta^-, \\ y^v - \sum_{k \in K^v_-} y^v_k & \text{if } \Delta^+ < \Delta^-. \end{cases}$$

Store the values $\{y^v_k\}$, for $k \in K^v$, $k \notin K^{v+1}$. Go to Step 1 with v increased to $v+1$.

Step 4. Retrieve the sequence $\{y^v_k\}$, for $v = 0, 1, 2 \ldots$. This constitutes an optimal solution.

This algorithm is finite, since at each iteration at least one optimal production quantity is determined. Optimality is not so easy to demonstrate. Bitran and Hax prove optimality by a careful analysis of the last iterations of the algorithm from which it can be established that the algorithm determines a solution satisfying the Kuhn–Tucker conditions.

7.3.5 THE ITEM-LEVEL SUBPROBLEMS

For a given item family k in product type j, one determines at this level production quantities of the individual items in the set N_k^{2j}.* The optimizations carried out at the product-type level and the item-family level determine in a sense the total production cost for the present planning period (variable production costs at the product-type level and setup costs at the item-family level). Nevertheless, there is the possibility of reducing costs in future periods by determining production quantities of the individual items in such a way that their individual runout times (defined as the production quantity plus the inventory on hand minus the safety stock, all divided by the demand forecast) come as close as possible to the runout time of the item family. If this is possible, these items would trigger more or less together in some later period, and they could hence be produced once more in one run at the cost of one setup (remember that all items within the same item family have similar seasonal demand patterns). Bitran and Hax propose an optimization problem that reflects this idea of minimizing the deviations in runout times. For each item family k in product type j they suggest

$$\text{Minimizing} \quad \sum_i \left(\frac{\tilde{x}_k^2 + \sum_i (\hat{I}_i^3 - \underline{I}_i^3)}{\sum_i r_i^3} - \frac{x_i^3 + \hat{I}_i^3 - \underline{I}_i^3}{r_i^3} \right)^2$$

$$\text{s.t.:} \quad \sum_i x_i^3 = \tilde{x}_k^2, \tag{7.10}$$

$$\underline{x}_i^3 \le x_i^3 \le \bar{x}_i^3 \ (i \in N_k^{2j}),$$

where all summations run over the index set N_k^{2j}. $\tilde{x}_k^2$ is the input from the item-family level. $\hat{I}_i^3, \underline{I}_i^3, \underline{x}_i^3$, and $\bar{x}_i^3$ are defined as in the discussion of the second-level optimization problem. r_i^3 is the demand forecast for item i for the present planning period. Again, r_i^3 has *not* been "netted" by deducting initial inventory. Problem (7.10) is again a convex optimization problem. Bitran and Hax describe a solution algorithm for (7.10), which is very similar to the one for solving (7.8). If (7.10) is not feasible, corresponding to case 1 or case 2 of the previous section, production quantities for individual items must be determined in some other manner.

7.3.6 A THREE-LEVEL SOLUTION PROCEDURE

The three-level procedure should now be clear. The LP problem (7.7) determines aggregate production levels for each product type over the entire

* Obviously, this is done only for item families to which positive production quantities are assigned in the item-family-level subproblems.

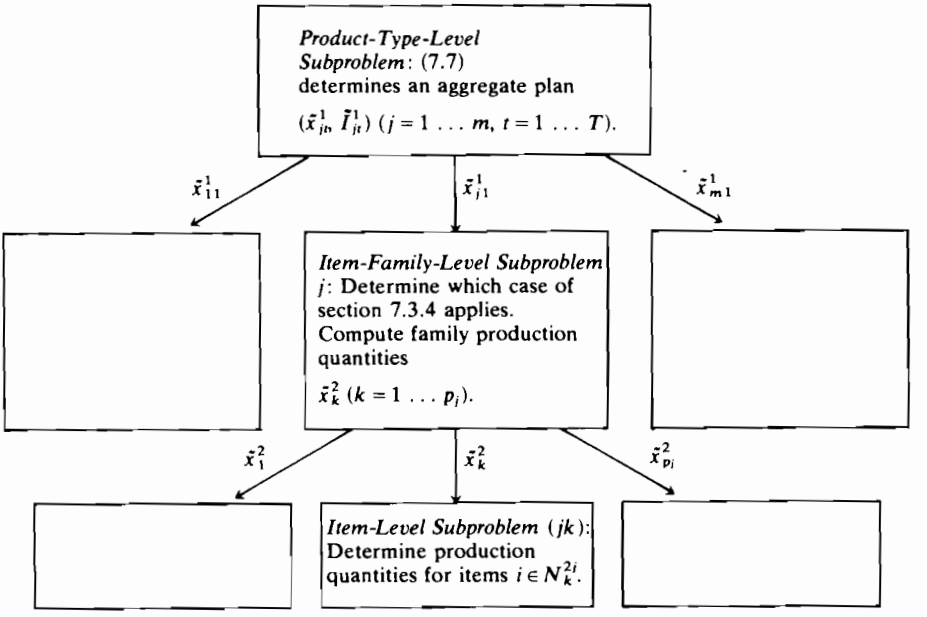


FIGURE 7.3 Hierarchical production planning: a three-level procedure.

planning horizon. The input to the m second-level subproblems, each corresponding to one product type, is the aggregate production quantity of the first period. For a given input, each item-family-level subproblem consists of finding production quantities for each family. Three cases can occur, the usual case requiring a solution to (7.8). The third-level (item-level) optimization, finally, determines individual item quantities within each given item family, with the second-level production quantity given as a parameter [problem (7.10)]. Figure 7.3 illustrates the procedure.

From this description two observations are called for. First, the meaning of an ultimate solution $\{\tilde{x}^3_i ; i \in N\}$ of the procedure in terms of feasibility for the original problem (7.1) is not clear. Second, the procedure is not multilevel in the sense of this volume since no interaction takes place between the various levels. We comment briefly on these issues.

Since the demand forecasts are derived independently at each level and discrepancies in the measured inventories at the various levels are allowed for as well, it should be clear that the ultimate solution $\{\tilde{x}^3_i ; i \in N\}$ need not even be feasible for the first period of the original problem. One might, nevertheless, pose an interesting theoretical question. Under what conditions is the third-level solution feasible for the original problem? The work of Gabbay (1975) deals with this question. If no discrepancies arise from disaggregating demand

and inventories from one level to another, and if, furthermore, only an "effective" requirements schedule is used on each subproblem level—i.e., if demand data for each item are adjusted for initial inventories—then Gabbay proves consistency of the hierarchical approach. That is, no back orders will be introduced by the disaggregation scheme, and a feasible solution obtains.

As for interaction between hierarchical levels, Bitran and Hax do, in fact, indicate that such a feature can be incorporated. That is, based on the solutions to the item-level subproblems (7.10), one may imagine some system for reallocating production quantities from one item family to another within the same product type. This idea could be incorporated in the solution process, and a "true" multilevel method would then result. It is questionable whether that would greatly enhance the practical significance of hierarchical production planning.

7.3.7 APPLICATIONS AND A COMPARISON WITH COLUMN GENERATION

As mentioned above, Hax and Meal (1975) report an implementation of a hierarchical production planning system. That system, however, is simpler than the one discussed here. Bitran and Hax (1976) have conducted some numerical experiments with the approach outlined above. These experiments lead to the conclusion that hierarchical production planning is computationally feasible and also efficient as a planning tool. Usefulness for real-life problems is difficult to evaluate, however, since no real-life application of the above approach has been reported.

Since the column generation approach of section 7.2 and the three-level method described in this section address the same problem types—i.e., problem (7.1)—a comparison is called for. First, we recall that both methods produce nonoptimal solutions to the original problem. Methodologically, the two approaches are completely different, and hierarchical production planning does not meet the requirement for "multilevelness' utilized in this volume. The key difference between the two methods is clearly the disaggregation scheme in hierarchical production planning. In the column generation approach, a detailed knowledge of the problem data for the entire planning horizon is required at the outset, which, with many thousands of products, may very well be unrealistic. In hierarchical production planning, it suffices to have information at the product-type level for the entire planning horizon. Operational issues are dealt with within a short planning period. As a consequence, the hierarchical approach is probably more robust with respect to forecasting errors (aggregate forecasts tend to be more correct). Also, the disaggregation scheme is attractive in production environments where there are many different products but where many of these products are very similar. All this suggests that a column generation approach would be most viable where the products are technologically distinct (making setup costs different) and where

reliable demand forecasts can be obtained. Where demand cannot be so easily estimated and where large classes of products are nearly identical, hierarchical production planning seems to be a more useful planning tool.

From a technical point of view, hierarchical production planning is probably easier to implement, since it uses ordinary linear programming and algorithms that can be programmed without great difficulties. Also, each subproblem can be replaced by more heuristic formulations, since it is really the disaggregation scheme that makes the method interesting. To implement a column generation algorithm may be more difficult, and there is little flexibility for adapting the method to the specifics of the problem situation.

REFERENCES

Armstrong, R., and A. C. Hax. 1974. A Hierarchical Approach for a Naval Tender Job Shop Design. Technical Report No. 101. Cambridge: Operations Research Center, Massachusetts Institute of Technology.

Bitran, G. R., and A. C. Hax. 1976. On the Design of Hierarchical Production Planning Systems. Technical Report. Cambridge: Operations Research Center, Massachusetts Institute of Technology.

Dzielinski, B. P., and R. Gomory. 1965. Optimal programming of lot sizes, inventory, and labor allocations. Management Science 11: 874–890.

Gabbay, H. 1975. A Hierarchical Approach to Production Planning. Technical Report No. 120. Cambridge: Operations Research Center, Massachusetts Institute of Technology.

Golovin, J. J. 1975. Hierarchical Integration of Planning and Control. Technical Report No. 116. Cambridge: Operations Research Center, Massachusetts Institute of Technology.

Gorenstein, S. 1970. Planning tire production. Management Science 17: B-72–B-82.

Hax, A. C., and H. C. Meal. 1975. Hierarchical integration of production planning and scheduling, pp. 53–69. In M. A. Geisler (ed.), Logistics. (TIMS Studies in the Management Sciences No. 1). Amsterdam: North-Holland.

Holt, C. C., F. Modigliani, J. F. Muth, and H. A. Simon. 1960. Planning Production, Inventories, and Work Force. Englewood Cliffs, New Jersey: Prentice-Hall.

Jones, C. H. 1967. Parametric production planning. Management Science 13: 843–866.

Lasdon, L. S., and R. C. Terjung. 1971. An efficient algorithm for multi-item scheduling. Operations Research 19: 949–969.

Manne, A. S. 1958. Programming of economic lot sizes. Management Science 4: 115–135.

Taubert, W. H. 1968. A search decision rule for the aggregate scheduling problem. Management Science 14: 343–359.

Wagner, H. M., and T. M. Whitin. 1958. A dynamic version of the economic lot size model. Management Science 5: 89–96.

8 Distribution Systems

8.1 INTRODUCTION AND OVERVIEW

In this chapter we report on two instances in which multilevel optimization techniques have been successfully implemented in distribution systems. By a distribution system we understand that part of an organization's logistical system that has to do with the delivery of produced output to final demand—the delivery, for example, of the produced commodities to the consumers in an industrial setting.

The two studies that we will discuss can be distinguished with respect to their overall goals. The first study, that of Geoffrion and Graves (1974), takes up the question of the optimal design of a distribution system. The central decision problem is to determine, at minimal distribution cost, a location pattern of distribution centers serving as the links between existing plants (each with a given production capacity for the given commodities) and the demand zones. Within each such zone, the demands for the various commodities are assumed to be known. The overall problem can be formulated as a mixed-integer linear programming problem and is solved by adapting the Benders algorithm, discussed in section 3.5. The Geoffrion–Graves model is a rather general version of what is known in the operations research literature as the plant-location model. Balinski and Spielberg (1969) offer an early discussion of such models. A study by Folie and Tiffin (1976) is reviewed in the second part of the chapter. Unlike Geoffrion and Graves, Folie and Tiffin concentrate on operational issues—that is, they attempt to determine an optimal production–distribution program for a given distribution system (somewhat different from the one in Geoffrion and Graves 1974). The problem is formulated as a minimal-cost multicommodity network flow problem. It is solved by a variant of the column generation scheme discussed in section 3.2 (the Ford–Fulkerson algorithm) in combination with generalized upper bounding.

8.2 THE OPTIMAL DESIGN OF A DISTRIBUTION SYSTEM

8.2.1 A MIXED-INTEGER PROGRAMMING FORMULATION

In the following discussion, it will be useful to let the index sets I, J, K, and L correspond to the set of commodities, the set of plants, the set of possible distribution center sites, and the set of demand zones. Generic elements are denoted by i, j, k, and l, respectively. Generally speaking, the goal is to choose from the set K some sites such that production at the plants can be channeled via those sites to satisfy demand in all demand zones, at the least distribution cost.

Let x_{ijkl} denote the amount of commodity i produced in factory j that is shipped to demand zone l by the distribution center k. If S_{ij} denotes the production capacity for product i at plant j, we can formulate the capacity constraints

$$\sum_{kl} x_{ijkl} \leq S_{ij} \qquad \text{for all } i \in I, j \in J.$$

To deal with the connection between distribution centers and demand zones, Geoffrion and Graves introduce the crucial assumption that each demand zone must be served by a single distribution center. If w_{kl} is a binary variable taking a value of 1 if site k serves zone l and zero otherwise, this assumption translates to

$$\sum_{k} w_{kl} = 1 \qquad \text{for all } l \in L.$$

Assuming that the known demands D_{il} have to be met, we can write

$$\sum_{j} x_{ijkl} = D_{il} w_{kl} \qquad \text{for all } i, k, l.$$

This can be interpreted to mean that whenever $w_{kl} = 0$, all flows x_{ijkl} must be zero. If $w_{kl} = 1$, all demand in zone l (for all commodities) has to be met via distribution center k.

With respect to the operations of distribution center k, a lower and upper bound on the annual throughput, $\underline{T}_k$ and $\bar{T}_k$, is presupposed. If v_k is a binary variable taking on a value of 1 if distribution center k is opened and zero otherwise, it follows that

$$\underline{T}_k v_k \leq \sum_{il} D_{il} w_{kl} \leq \bar{T}_k v_k \qquad \text{for all } k.$$

The quantity $\sum_{il} D_{il} w_{kl}$ measures the throughput of distribution center k and is required to be zero if $v_k = 0$ and to be within capacity bounds if $v_k = 1$.

Geoffrion and Graves also allow the possibility of including additional linear constraints on the w_{kl} and v_k variables. This enhances the scope of the model without changing the solution methods appreciably (see Geoffrion and Graves 1974, p. 825). Since we are mainly interested in the methodological aspects, we will not include such additional constraints.

As for the cost structure, linear transportation costs at a unit price of c_{ijkl} are assumed. To account for the operation of a distribution center, a fixed-cost portion f_k is included if center k is opened, as is a linear part with a unit cost of g_k.

One then obtains the following mixed-integer programming problem, the overall problem under consideration:

$$\text{Minimize} \quad \sum_{ijkl} c_{ijkl}x_{ijkl} + \sum_{k}\left\{f_k v_k + g_k\left(\sum_{il} D_{il}w_{kl}\right)\right\}$$

$$\text{s.t.:} \quad \sum_{kl} x_{ijkl} + s_{ij} = S_{ij} \qquad (i \in I, j \in J), \tag{8.1a}$$

$$\sum_{j} x_{ijkl} = D_{il}w_{kl} \qquad (i \in I, k \in K, l \in L), \tag{8.1b}$$

$$\sum_{k} w_{kl} = 1 \qquad (l \in L), \tag{8.1c}$$

$$\underline{T}_k v_k \leq \sum_{il} D_{il}w_{kl} \leq \bar{T}_k v_k \qquad (k \in K), \tag{8.1d}$$

$$\text{all } v_k \text{ and } w_{kl} = 0 \text{ or } 1, \tag{8.1e}$$

$$\text{all } x_{ijkl} \geq 0,\ s_{ij} \geq 0.$$

Slack variables s_{ij} have been introduced in (8.1a). Several features of the overall problem (8.1) are discussed in Geoffrion and Graves (1974, pp. 823–826). It is sufficient to note here that (8.1) could be a very large problem and hence not easily solvable by single-level techniques. Geoffrion and Graves therefore proposed an application of Benders decomposition.

8.2.2 APPLICATION OF THE BENDERS ALGORITHM

In view of the developments in section 3.5, the specialization of the Benders algorithm to the overall problem (8.1) is reasonably straightforward. The variables x_{ijkl} and s_{ij} are the "linear" ones in formulation (3.23) of Chapter 3. The vector of "special" variables y in (3.23) corresponds to the vector $(v, w) = (v_k, w_{kl};\ k \in K, l \in L)$. The form of $f(y)$ is obvious from the objective function in (8.1). The constraints $Ax + F(y) \geq b$ correspond to (8.1a)–(8.1b), and the set Y corresponds to the set of vectors $y = (v, w)$ satisfying (8.1c)–(8.1e). There is a minor difference in that (8.1a) and (8.1b) are strict equalities,

as opposed to the inequality $Ax + F(y) \geq b$ in the general formulation. This, however, causes no difficulties.

In the earlier discussion in section 3.5, special care was taken to handle certain infeasibilities. That is, if the special variables are fixed, an ordinary LP problem results. This problem, however, could be infeasible because of unfortunate choices of the special variables. To eliminate such infeasibilities, restrictions

$$\tilde{u}^r(b - F(y)) \leq 0 \qquad (r = 1 \ldots R)$$

were incorporated in formulation (3.28) of Chapter 3. In this case, it is assumed that for any choice of the binary variables satisfying (8.1c)–(8.1e) of problem (8.1) above, there will always be a feasible and bounded choice of the linear variables. This means, in particular, that $\sum_j S_{ij} \geq \sum_l D_{il}$ for all i. There will hence be no constraints of the form $\tilde{u}^r(b - F(y)) \leq 0$ in the supremal subproblem in the present case.

Now suppose the binary variables v_k and w_{kl} are fixed to $\bar{v}_k$ and $\bar{w}_{kl}$ satisfying (8.1c)–(8.1e). The following LP problem results:

$$\begin{aligned}
\text{Minimize} \quad & \sum_{ijkl} c_{ijkl} x_{ijkl} \\
\text{s.t.:} \quad & \sum_{kl} x_{ijkl} + s_{ij} = S_{ij} \qquad (i \in I, j \in J), \\
& \sum_{j} x_{ijkl} = D_{il} \bar{w}_{kl} \qquad (i \in I, k \in K, l \in L), \\
& x_{ijkl} \geq 0, \; s_{ij} \geq 0.
\end{aligned} \tag{8.2}$$

Following the development of section 3.5, we would solve the dual of (8.2) as an infimal subproblem. However, one may, of course, equally well solve the primal problem (8.2) and then simply recover the optimal dual solution. We now note that (8.2) separates completely into independent transportation problems, one for each commodity i, of the following form:

$$\begin{aligned}
\text{Minimize} \quad & \sum_{jkl} c_{ijkl} x_{ijkl} \\
\text{s.t.:} \quad & \sum_{kl} x_{ijkl} + s_{ij} = S_{ij} \qquad (j \in J), \\
& \sum_{j} x_{ijkl} = D_{il} \bar{w}_{kl} \qquad (k \in K, l \in L), \\
& x_{ijkl} \geq 0, \; s_{ij} \geq 0.
\end{aligned} \tag{8.3}$$

Hence, one has not one, but N, infimal subproblems, where N is the number of commodities (the cardinality of I). In (8.3), the sources are the factories.

There is one destination for each combination of k and l. The delivery requirements of most of the destinations are zero, however. This follows since, for each l, $\bar{w}_{kl} = 1$ only for one k, and $\bar{w}_{kl} = 0$ for the rest.

Let the optimal dual multipliers associated with (8.3) be denoted μ_{ij} and π_{ikl}. To evaluate these multipliers, one may add a slack destination to (8.3) and set the corresponding multiplier equal to zero, and then successively obtain each μ_{ij} and π_{ikl}. Actually, Geoffrion and Graves derive the optimal dual multipliers in a more efficient fashion; see Geoffrion and Graves (1974, pp. 828–830) for details.

Figure 8.1 depicts the adjustment phase of the Benders algorithm as applied to the overall problem (8.1). This figure, together with the discussion in section 3.5.1, should allow the reader to reconstruct the details of the algorithm. The supremal subproblem involves certain binary variables, which may be considered to represent investment decisions. The infimal subproblems are transportation problems, each pertaining to one specific commodity.

8.2.3 THE IMPLEMENTATION OF GEOFFRION AND GRAVES

The specific variant of the Benders decomposition actually used by Geoffrion and Graves differs a little from the algorithm displayed in Figure 8.1. The supremal subproblem that they solve in each iteration t is the following: Find a feasible solution to the restrictions:

$$\begin{aligned}
&\sum_k \left\{ f_k v_k + g_k\left(\sum_{il} D_{il} w_{kl}\right)\right\} \\
&+\sum_{ij} \mu_{ij}^p S_{ij} + \sum_{ikl} \pi_{ikl}^p D_{il} w_{kl} \leq U - \varepsilon \quad (p \in \mathcal{P}), \\
&\sum_k w_{kl} = 1 \qquad (l \in L), \qquad (8.4) \\
&\underline{T}_k v_k \leq \sum_{il} D_{il} w_{kl} \leq \bar{T}_k v_k \qquad (k \in K), \\
&v_k,\ w_{kl} = 0 \text{ or } 1.
\end{aligned}$$

In this formulation (8.4), U is the best upper bound on the optimal solution value of the overall problem (8.1) obtained so far (see the discussion in section 3.5.1 of upper and lower bounds). ε is a positive tolerance level. A feasible solution to (8.4) (if one exists) can be found by specifying an arbitrary linear objective function and then optimizing the resulting 0–1 integer programming problem. Geoffrion and Graves actually used the last constraint added to the supremal subproblem as the objective function.

If no feasible solution to the supremal subproblem exists, the adjustment phase terminates, and a solution to the original problem (8.1) may be

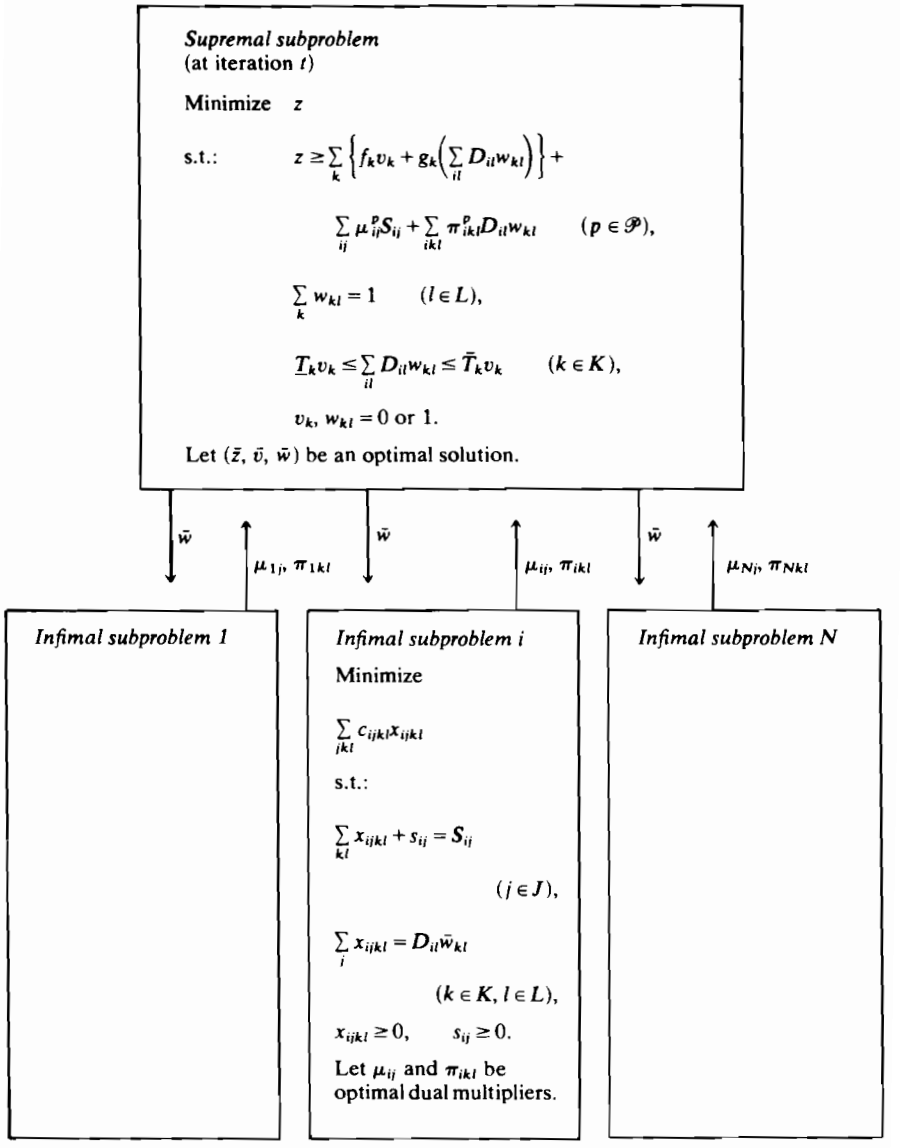


FIGURE 8.1 The adjustment phase of the Benders algorithm for distribution system design.

TABLE 8.1 Performance of Benders Algorithm for Distribution System Design Problem

Number of Distribution Centers	Number of Binary Variables	Number of Rows	Tolerance Level (as % of Optimal Cost)	Number of Iterations
16	249	4403	0.06	3
16	254	4488	0.03	4
18	287	4944	0.03	5
19	336	5657	0.06	4
21	349	5783	0.15	4
25	411	6857	0.06	7
25	411	6837	0.15	4
26	427	7054	0.15	5
31	513	8441	0.15	5

SOURCE: Geoffrion and Graves (1974, p. 837).

recovered.* It follows from the construction of this variant that it does not necessarily yield an optimal solution. It must, however, converge to an ε-optimal (near-optimal) solution in a finite number of iterations.

The above variant has certain computational advantages, among others that the supremal subproblem (8.4) is a *pure* 0–1 problem, whereas the supremal subproblem in Figure 8.1 involves one continuous variable, namely z.

The successive supremal subproblems were solved in the Geoffrion–Graves study by a combination of the branch and bound technique and Gomory's cutting-plane method. The subroutine for the cutting-plane method utilized generalized upper bounding for the constraints $\sum w_{kl} = 1$ as well.

To illustrate the impressive convergence properties of the resulting algorithm, we give some results in Table 8.1, where it can be seen that near-optimal solutions to a number of test problems could be obtained in remarkably few iterations.

The method was applied to a real-life problem arising in the food industry. The problem had 11,854 rows, 727 binary variables, and 23,513 continuous variables. The paper of Geoffrion and Graves contains interesting discussions of the various analyses carried out with the aid of this adaptation of the Benders algorithm.

* The resulting solution to the overall problem (8.1) may be recovered in the execution phase as follows: The supremal subproblem solution of the next to last iteration of the adjustment phase provides the v_k and w_{kl} components. Given those w_{kl}, one solves the resulting infimal subproblems (the transportation problems), to obtain the x_{ijkl} components of the overall solution.

8.3 DETERMINING OPTIMAL PRODUCTION–DISTRIBUTION PROGRAMS

8.3.1 A NETWORK FLOW FORMULATION

In the study of Folie and Tiffin (1976), the focus is, as already mentioned, on operational issues. Moreover, they do not attempt to solve large-scale problems of the type discussed in the previous section. Rather, their goal is to develop a method that can be implemented on commercial computer systems, and whose output is easily interpretable. Again, the underlying real-life problem stems from the food industry. Here we will present the model of Folie and Tiffin using the notation of section 8.2, so as to facilitate a comparison with the Geoffrion–Graves model. The presentation in the original paper is somewhat different. The particular implementation is briefly discussed in section 8.3.3.

The index sets I, J, K, and L correspond to the set of commodities produced by the firm, the set of plants of the firm, the set of distribution centers operated by the firm, and the set of demand zones (defined as regional warehouses). Each plant j has a productive capacity for commodity i given by S_{ij}, as well as an overall production capacity of S_j. The production costs are linear with a unit cost of c_{ij}.* The model also has linear transportation costs, with c_{ijk} being the unit cost of shipping commodity i produced in plant j to distribution center k, and with c_{ikl} the unit cost of shipping commodity i from distribution center k to demand zone l. The unit cost of delivering commodity i to demand zone l is hence dependent on the plant where the commodity was produced as well as on the particular transportation route, or

$$c_{ijkl} = c_{ij} + c_{ijk} + c_{ikl}.$$

The objective then becomes to find flows $\{x_{ijkl} | i \in I, j \in J, k \in K, l \in L\}$ within the given capacity bounds such that the final demand $\{D_{il} | i \in I, l \in L\}$ is met. This may be written as:

$$\text{Minimize} \quad \sum_{ijkl} c_{ijkl} x_{ijkl}$$

$$\text{s.t.:} \quad \sum_{kl} x_{ijkl} \leq S_{ij} \qquad (i \in I, j \in J), \tag{8.5a}$$

$$\sum_{ikl} x_{ijkl} \leq S_j \qquad (j \in J), \tag{8.5b}$$

$$\sum_{jk} x_{ijkl} = D_{il} \qquad (i \in I, l \in L), \tag{8.5c}$$

$$\text{all } x_{ijkl} \geq 0.$$

* Infeasible combinations can be ruled out by setting costs sufficiently high.

Problem (8.5) is the overall problem in this discussion. A comparison with (8.1), the overall problem of Geoffrion and Graves, reveals certain differences. The design aspects (modeled through binary variables) present in (8.1) are missing in (8.5). But even if a feasible choice of the binary variables in (8.1) were given, the two programs do not coincide. Folie and Tiffin do not assume that each demand zone must be served by a single distribution center, which explains the difference between (8.1*b*) and (8.5*c*). The overall production capacity constraints (8.5*b*) are missing in (8.1). Problem (8.5) is a minimal-cost multicommodity network flow problem. It is closely related to the maximal multicommodity network flow problem discussed in section 3.2.

The particular application of Folie and Tiffin cannot be said to be a large-scale one. Nonetheless, one can easily imagine that the overall problem (8.5) could be so large that straightforward single-level methods become inapplicable. One must then resort to sophisticated single-level methods of the kind discussed in Maier (1974) (based on the equivalent node–arc formulation), or two-level methods of the kind described below.

8.3.2 A COLUMN GENERATION METHOD

The principle of column generation was outlined in section 3.2.1. Application to the overall problem (8.5) is straightforward. Just as in the discussion of the maximal multicommodity network flow problem in section 3.2.3, there will be one infimal subproblem for each commodity. If there are N commodities, there will hence be N infimal subproblems.* The supremal subproblem will be of the same type as (8.5), but incorporating only a subset of the columns.

Now suppose the supremal subproblem has been solved in some iteration of the two-level method. The question is then whether it is profitable to add additional columns to the supremal subproblem. This question is resolved by considering for each commodity the following infimal subproblem:

$$\begin{aligned} &\text{Minimize} \quad c_{ijkl} - \mu_{ij} - \lambda_j - \pi_{il} \\ &\text{over the sets } J, K, \text{ and } L. \end{aligned} \tag{8.6}$$

μ_{ij}, λ_j, and π_{il} are the dual multipliers associated with constraints (8.5*a*), (8.5*b*), and (8.5*c*) of the supremal subproblem in the current iteration.

Problem (8.6) turns out to be a very simple shortest-path problem. Consider a directed network with node set consisting of J, K, and L. The costs of traversing an arc are given as

$$c^i_{jk} = c_{ij} + c_{ijk} - \mu_{ij} - \lambda_j \qquad \text{for } j \in J, k \in K;$$

$$c^i_{kl} = c_{ikl} - \pi_{il} \qquad \text{for } k \in K, l \in L.$$

* One could also have one infimal subproblem for each product–demand zone combination (this seems to be what Folie and Tiffin used).

It is obvious that finding the shortest path from the set of sources (the set of plants) to the set of sinks (the set of demand zones) is equivalent to solving (8.6). If, for each commodity i, the shortest path has nonnegative length, no new column is added to the supremal subproblem, meaning that the last supremal subproblem solution is optimal for the overall problem (8.5). If, for some commodity i', the shortest path has negative length, a corresponding column is added to the supremal subproblem. That column has three elements equal to unity, and the rest equal to zero. If the shortest path traverses nodes j', k', and l', there will be a 1 in that constraint (8.5a) for which $i = i'$, $j = j'$; a 1 in that constraint (8.5b) for which $j = j'$; and a 1 in that constraint (8.5c) for which $i = i'$, $l = l'$. The objective function coefficient is $c_{ijkl} = c_{i'j'} + c_{i'j'k'} + c_{i'k'l'}$.

8.3.3 THE IMPLEMENTATION OF FOLIE AND TIFFIN

The column generation method described above was programmed and compared with ordinary LP in solving some small test problems. Generalized upper bounding was used to deal with the constraints (8.5c). The single-level LP solution method was applied, not to problem (8.5) directly, but to the equivalent node–arc formulation. The two-level method performed substantially better than the single-level method with regards to computing time and number of iterations both with and without the generalized upper bounding feature (Folie and Tiffin 1976, pp. 293–294).

In the practical application problem, there were nine commodities, eight plants, four distribution centers, and four demand zones. This results in a reasonably small overall problem (8.5). That overall problem could, in fact, have been explicitly generated by a matrix generation routine and would have been within reach of ordinary LP. However, column generation plus generalized upper bounding was used. An ICL 1902A computer was utilized. Folie and Tiffin report the successful use of the method to resolve various planning problems in the company.

Finally, it may be mentioned that an entirely different two-level method for solving minimal-cost multicommodity network flow problems is discussed by Kennington and Shalaby (1977).

REFERENCES

Balinski, M. L., and K. Spielberg. 1969. Methods for integer programming: algebraic, combinatorial, and enumerative, pp. 195–292. In J. S. Aronofsky (ed.), Progress in Operations Research. Vol. III. New York: Wiley.

Folie, M., and J. Tiffin. 1976. Solution of a multi-product manufacturing and distribution problem. Management Science 23: 286–296.

Geoffrion, A. M., and G. W. Graves. 1974. Multicommodity distribution system design by Benders decomposition. Management Science 20: 822–844.

Kennington, J., and M. Shalaby. 1977. An effective subgradient procedure for minimal cost multicommodity flow problems. Management Science 23: 994–1004.

Maier, S. F. 1974. A compact inverse scheme applied to a multi-commodity network with resource constraints, pp. 179–203. In R. W. Cottle and J. Krarup (ed.), Optimization Methods for Resource Allocation. London: The English Universities Press.

9 Freight Ship Route Scheduling and Electricity Generation

9.1 INTRODUCTION AND OVERVIEW

In this chapter we will discuss two studies that could not be conveniently grouped in the other chapters. The first study is concerned with the derivation of optimal ship itineraries for a shipping company. An itinerary is a sequence of cargoes. The company owns a number of ships, characterized by size, cruising speed, initial position, and the like. Each ship can handle a given set of cargoes, each characterized by size, loading dates, origin, and destination. The discussion is based on the work of Appelgren (1969, 1971). The problem was originally formulated as a network in its arc–chain form, and column generation was applied to it. To overcome difficulties associated with the occurrence of fractional solutions, column generation was combined with a branch and bound method. This means that the algorithm finally implemented by a Swedish shipping company is, in fact, a three-level method. The ship route scheduling problem is discussed in section 9.2.

In section 9.3 we review a study of optimal electricity generation (Chaly *et al.* 1974). For a given power system, one wants to generate electricity at minimal fuel cost within given capacity limits, while satisfying demand for electricity (power losses are explicitly included). The resulting convex programming problem can be solved by the nonlinear Dantzig–Wolfe decomposition method.

9.2 FREIGHT SHIP ROUTE SCHEDULING

9.2.1 PROBLEM FORMULATION

Consider a shipping company owning I ships, indexed by $i = 1 \ldots I$. Suppose the company has the opportunity to handle K cargoes, indexed by $k = 1 \ldots K$.

For some planning period (e.g., the next 60 days), the company must decide on an itinerary for each ship. An itinerary is characterized by a sequence of cargoes. Depending on the set of available cargoes, and the initial position of ship i, its size, speed characteristics, and other factors, there is a particular set of feasible itineraries for that ship. Suppose there are altogether $N(i)$ feasible itineraries for ship i, indexed by j (to remain idle is always one feasible itinerary). Associated with each itinerary j is a payoff, denoted v_{ij}. The objective function proposed in Appelgren (1969) is then to maximize the sum of the payoffs for all ships:

$$\text{Maximize} \quad \sum_{i} \sum_{j=1}^{N(i)} v_{ij} x_{ij},$$

where x_{ij} is a binary variable equal to 1 if ship i takes itinerary j and equal to 0 otherwise. Since one ship can, by definition, take only one itinerary during the planning period, one obtains the constraints

$$\sum_{j=1}^{N(i)} x_{ij} = 1 \quad (i = 1 \ldots I).$$

Also, any cargo can be carried at most once. Define $a_{ijk} = 1$ if cargo k is taken up in itinerary j of ship i; otherwise $a_{ijk} = 0$. It must then hold that

$$\sum_{i} \sum_{j=1}^{N(i)} a_{ijk} x_{ij} \left\{ \begin{matrix} = 1 \\ \leq 1 \end{matrix} \right\} \quad (k = 1 \ldots K).$$

The equality restrictions refer to those cargoes for which the company has entered into a contractual obligation. However, there may also be some optional cargoes that can be picked up if the shipping company so decides, hence the restrictions in inequality form.

One obtains the following integer programming problem:

$$\text{Maximize} \quad \sum_{i=1}^{I} \sum_{j=1}^{N(i)} v_{ij} x_{ij}$$

$$\text{s.t.:} \quad \sum_{j=1}^{N(i)} x_{ij} = 1 \qquad (i = 1 \ldots I), \tag{9.1a}$$

$$\sum_{i=1}^{I} \sum_{j=1}^{N(i)} a_{ijk} x_{ij} \left\{ \begin{matrix} = 1 \\ \leq 1 \end{matrix} \right\} \qquad (k = 1 \ldots K), \tag{9.1b}$$

$$\text{all } x_{ij} = 0 \text{ or } 1.$$

Problem (9.1) is the overall problem of this section. It could involve 40 ships and 50 cargoes, and the number of itineraries could hence be very large, rendering the application of straightforward integer programming techniques impossible.

Appelgren (1969) suggested solving (9.1) by a two-level procedure based on the idea of dropping the integrality conditions and on the application of column generation to the resulting LP problem. It was hoped for that the LP solution to (9.1) would be "almost" integer, i.e., that it would be such that almost all ships get assigned to precisely one itinerary (and not a mixture of two or more). The remaining ships could then be rescheduled by manual methods. Subsequently, Appelgren (1971) described a three-level method that produces integer-valued solutions. The intermediate and infimal subproblems correspond to the supremal and infimal subproblems in the two-level method of Appelgren (1969). The supremal subproblem determines iteratively which fractional variables are set to which integral values. The method combines a branch-and-bound method with column generation. This method was implemented in a Swedish shipping company. Since the generation of ship itineraries is of central interest, a detailed discussion is given in section 9.2.2. The two-level method is described separately in section 9.2.3, which makes the subsequent discussion of the three-level method in section 9.2.4 easier.

We note that, disregarding the integrality constraints, (9.1) is a multicommodity network flow problem (each ship is a commodity).

9.2.2 THE GENERATION OF SHIP ITINERARIES

The construction of itineraries will now be considered in detail. A cargo is obviously characterized by a port of origin and a destination. It is, moreover, characterized by one or several alternative loading dates. That is, a cargo may be available on one or several alternative loading dates, for instance Monday through Friday of a particular week.

For each ship, a network representation of the available itineraries can be used. To be concrete, suppose there are altogether three cargoes available to the shipping company, each with two alternative loading dates. Now consider some particular ship i. Whether a particular itinerary is feasible for that ship depends on several things, such as initial position and ship size. In the present case, suppose that the following one-cargo itineraries are all feasible for the ship under consideration: (1, 1), (1, 2), (2, 1), (2, 2), and (3, 2). In this notation, the first index represents cargo, and the second, loading date alternative. The one-cargo itinerary (3, 1) is assumed not feasible, for instance, because the current position of the ship is so far away from the origin harbor of cargo 3 that it is impossible to reach that harbor by the date given by loading date alternative 1. Suppose, furthermore, that the following two-cargo itineraries are also feasible: (1, 1)–(2, 1); (1, 1)–(2, 2); (1, 2)–(2, 2). No three-cargo sequence is feasible (for instance, because of the crusing speed of the ship under consideration). Now order all the cargo-loading date combinations according to increasing loading dates. Suppose that order is (3, 1), (3, 2), (1, 1), (1, 2), (2, 1), (2, 2) in the above example case. This means that the

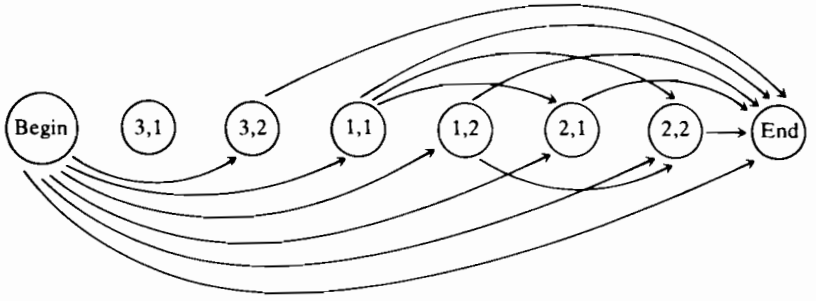


FIGURE 9.1 Example network of ship itineraries.

second loading date alternative for cargo 3 occurs earlier than the first loading date alternative for cargo 1. The set of all feasible itineraries for the particular ship under consideration can then be represented as a network, where the nodes signify cargo-loading date combinations, as depicted in Figure 9.1. Note that the nodes are ordered according to increasing time on a time axis. Any path from "Begin" to "End" represents a feasible itinerary. The arc directly from "Begin" to "End" is the idle alternative. There are altogether nine different itineraries, including the idle one.

Associated with each arc in the network is a payoff element. The payoff from the arc between "Begin" and (1, 2), for instance, is the payoff resulting from letting cargo 1, loaded on the second loading date alternative, be the first cargo in the itinerary for the ship under consideration. That payoff includes the cargo revenue minus cruising costs, which also means the cost of cruising empty from the current position to the origin port of the cargo. The payoff from the arc between (1, 2) and (2, 2) consists of the revenue from carrying the second cargo minus cruising costs between the destination harbor of cargo 1 and the origin harbor of cargo 2 (which could be zero, if these are the same harbor), possible idle time costs while waiting for the second loading time alternative for cargo 2, and cruising costs for carrying cargo 2. The payoff from the arc between (2, 2) and "End" could include the cost of waiting idle in the destination port of cargo 2 for the rest of the planning period. The total payoff of ship i from the jth itinerary, v_{ij}, is hence equal to the sum of partial payoffs on the arcs of the relevant path in the network. So, if the directed arc $((q, m), (k, n))$ belongs to a feasible itinerary for ship i, where q and k represent subsequent cargoes and m and n loading dates, the partial payoff can be written as v_{iqmkn}. If $q = m = 0$, then k is the first cargo on the itinerary. If $k = n = 0$, then q is the last cargo on the itinerary. The idle itinerary is identified when $q = m = k = n = 0$. Define corresponding zero–one variables x_{iqmkn}. If $x_{iqmkn} = 1$, then the arc$((q, m), (k, n))$ is on a given itinerary. If $x_{i00kn} = 1$, then "Begin"–(k, n) is on the itinerary, meaning that k is the first cargo. If $x_{iqm00} = 1$, then (q, m)–"End" is on the itinerary, in which case q is the last cargo. If $x_{i0000} = 1$, then the

itinerary is the idle one. The two-cargo itinerary (1, 2)–(2, 2) would hence be defined by $x_{i0012} = x_{i1222} = x_{i2200} = 1$, and all other $x_{iqmkn} = 0$. The corresponding total payoff $v_{ij} = v_{i0012} + v_{i1222} + v_{i2200}$. The one-cargo itinerary (1, 2) would be defined by $x_{i0012} = x_{i1200} = 1$.

Hence, all feasible ship itineraries for a given vessel can be generated by the construction of networks like that illustrated in Figure 9.1. The total payoff of an itinerary can be reconstructed from the partial payoffs associated with the arcs in the network. Every feasible itinerary can be characterized by a set of binary variables.

9.2.3 A COLUMN GENERATION SCHEME

We will now consider a column generation scheme for the ship scheduling problem. Let $\pi = (\pi_1, \pi_2 \ldots \pi_K)$ be the simplex multipliers associated with the cargo constraints (9.1*b*). Let δ_i $(i = 1 \ldots I)$ be the multiplier associated with the ith ship constraint (9.1*a*). A new itinerary for ship i would be represented in the supremal subproblem by a column vector with $(I + K)$ elements. Out of the first I elements, the ith will be 1 (corresponding to the relevant constraint (1*a*)), and the remaining elements zero. The last K elements will either be 0 or 1, depending on which cargoes are picked up in the itinerary. The coefficient in the objective function will, of course, be v_{ij}. To determine whether there is any worthwhile itinerary to be added to the ones already at hand for ship i, one would maximize $v_{ij} - \sum_{k=1}^{K} a_{ijk}\pi_k - \delta_i$ over all feasible itineraries. If this quantity is positive, then a new column for the supremal subproblem has been identified.

The problem of maximizing $(v_{ij} - \sum_{k=1}^{K} a_{ijk}\pi_k - \delta_i)$ over all feasible itineraries may also be written as

$$
\begin{aligned}
\text{Maximize} \quad & \sum_{qmkn} (v_{iqmkn} - \pi_k) x_{iqmkn} - \delta_i \\
\text{s.t.:} \quad & \sum_{kn} x_{iqmkn} - \sum_{kn} x_{iknqm} = 0 \\
& (\text{all } q, m;\ q \neq 0), \qquad (9.2) \\
& \sum_{kn} x_{i00kn} = 1, \\
& \text{all } x_{iqmkn} = 0 \text{ or } 1.
\end{aligned}
$$

For $k = 0$, π_k is defined to be zero. The constraints in (9.2) describe the requirement that the solution be a feasible itinerary. Problem (9.2) corresponds to finding the longest path through a network. It can be solved efficiently by a simple backward recursion since the underlying network is acyclic, as is evident from Figure 9.1. From the structure of the network in

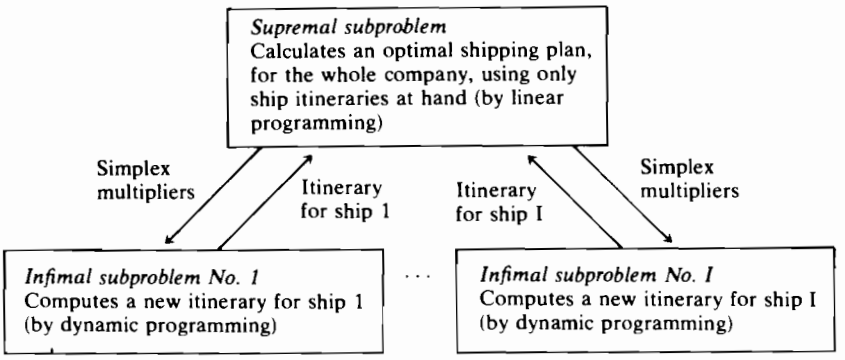


FIGURE 9.2 The adjustment phase of the two-level method for ship scheduling.

Figure 9.1, it is also clear that the necessary arcs of the network can be constructed while performing the backward recursion (this construction is simplified if the nodes of the network are ordered according to increasing loading date). The resulting two-level method is straightforward, and is clarified in Figure 9.2.

Appelgren (1969) successfully solved some test problems using column generation. One typical, realistic problem involved 40 ships, 50 cargoes, and a 60-day planning period. Solution time was about 2.5 minutes on an IBM 7090 computer. A somewhat simplified version of the same problem was also solved in a single-level fashion, by ordinary linear programming. This meant that all feasible itineraries for each ship had to be generated in advance. Actually, this was not done; only a subset was generated (this is the simplification just referred to). Solution time was now around 20 minutes using a standard LP code. The ship scheduling problem considered here hence lends itself to quite successful applications of a two-level method. It may be remarked, though, that these experiments were carried out some 10 years ago, and today the relative advantage of the two-level method may be smaller, since better standard LP codes are now available.

The solution obtained by this two-level method need not be integer-valued. To obtain some insight into the occurrence of fractional solutions, Appelgren conducted a series of experiments. He concluded that the frequency of fractional solutions was about 1–2 percent for randomly generated test problems (for details, see Appelgren 1969, pp. 63–68). This induced Appelgren to combine the present approach with integer programming methods, with a three-level method as the result.

9.2.4 A THREE-LEVEL METHOD

Here we will just outline the method developed in Appelgren (1971); some details will not be given, since their inclusion would make the discussion somewhat technical.

Appelgren's method combines column generation with a branch-and-bound method originally developed by Land and Doig (1960). The basic principle is as follows. Suppose one has obtained, by means of any algorithm, a fractional solution to some 0–1 linear integer programming problem. Then one selects some column from the constraint matrix corresponding to a fractional variable, and sets the corresponding variable respectively equal to zero (forcing the column out of the previously obtained solution) and equal to one (forcing the column in). The given algorithm is applied to the two resulting (more constrained) problems, each constituting the first nodes of separate "branches" of the "tree." This procedure can be continued, assuming an integer solution was obtained in some branch, until the solution value of the best integral solution is not exceeded by any fractional bound that is constructed at any stage of the search procedure.*

The three-level method then proceeds in the following manner: If the column generation method (accounting for the two lower levels) produces a fractional solution, the supremal subproblem decides which fractional value to branch on. The two restricted problems that result are then solved by the lower levels. The procedure continues until an optimality test at the supremal level is passed; this occurs as soon as the upper bounds obtained from the fractional solutions at the deepest nodes of the tree are smaller than or equal to the solution value of some previously generated integer solution.

Appelgren developed various selection rules to determine the next variable to branch on. These rules are described in detail in Appelgren (1971, pp. 68–69). A technical problem arises from the fact that the column generation method may very well generate columns that have been ruled out by the branching procedure. This difficult problem was solved in an *ad hoc* fashion (see the discussion in Appelgren 1971, pp. 70–71).

This three-level method was implemented in a Swedish shipping company. In fact, it has been used once or twice a week since late 1970. Typical problems involve 100 ships and 135 cargoes. Appelgren reports that the computer-produced schedules must generally be somewhat revised manually but that they are, nevertheless, valuable as tentative plans (Appelgren 1971, p. 77).

* Once an integer solution is obtained in a branch, no further iterations are necessary in that branch.

9.3 PLANNING POWER GENERATION

9.3.1 PROBLEM FORMULATION

Chaly *et al.* (1974) formulate the problem of planning for power generation in a system with hydroelectric as well as thermal power stations as a convex programming problem. It turns out that for power systems containing only thermal power stations, the application of the Dantzig–Wolfe decomposition principle is computationally attractive.

Assume a power network with M nodes. At the first N nodes ($N \leq M$) there are load (i.e., demand) points as well as thermal power stations. In the remaining $M-N$ nodes, there are only load points. For each power station $i = 1 \ldots N$, x_i represents the power generated at some point in time under consideration. Let $\phi_i(x_i)$ be the fuel usage at station i associated with a production x_i. Chaly *et al.* propose fuel-use minimization over the entire network as the overall objective:

$$\text{Minimize} \quad \sum_{i=1}^{N} \phi_i(x_i).$$

Each ϕ_i is assumed to be convex.

This minimization takes place subject to certain constraints. Let there be K power lines in the network. The following restrictions express upper and lower bounds on the power flow of each line in terms of the power generated at the individual stations:

$$\sum_{i=1}^{N} a_{ik} x_i \leq U_k \qquad (k = 1 \ldots K),$$

$$-\sum_{i=1}^{N} a_{ik} x_i \leq -L_k \qquad (k = 1 \ldots K).$$

There are capacity constraints on the stations as well:

$$\underline{x}_i \leq x_i \leq \bar{x}_i \qquad (i = 1 \ldots N).$$

Additionally, there is a constraint on the power balance in the network. Let $d = (d_1 \ldots d_M)$ give the load at the different nodes of the network. There is a power loss function $F(x, d)$, $x = (x_1 \ldots x_N)$, identifying the losses occuring in the network for given amounts of power generated and demanded at the different nodes. The function $F(.\,,.)$ is assumed to be convex. Since the demand schedule has to be met, one obtains

$$\sum_{i=1}^{N} x_i - F(x, d) \geq \sum_{i=1}^{M} d_i.$$

Thus, the following convex programming problem has to be solved:

$$\begin{aligned}
\text{Minimize} \quad & \sum_{i=1}^{N} \phi_i(x_i) \\
& \sum_{i=1}^{N} a_{ik} x_i \leq U_k \qquad (k = 1 \ldots K), \\
& -\sum_{i=1}^{N} a_{ik} x_i \leq -L_k \qquad (k = 1 \ldots K), \\
& \underline{x}_i \leq x_i \leq \bar{x}_i \qquad (i = 1 \ldots N), \\
& \sum_{i=1}^{N} x_i - F(x, d) \geq \sum_{i=1}^{M} d_i.
\end{aligned} \tag{9.3}$$

Problem (9.3) was solved by Chaly *et al.* using Dantzig–Wolfe decomposition for nonlinear programs. It is the original, or overall, problem of this section.

9.3.2 APPLICATION OF THE DANTZIG–WOLFE METHOD

The discussion on this section is based on section 3.4. First define

$$X = \left\{ x \,|\, \underline{x}_i \leq x_i \leq \bar{x}_i \ (i = 1 \ldots N), \sum_{i=1}^{N} x_i - F(x, d) \geq \sum_{i=1}^{M} d_i \right\}.$$

The set of power schedules X is convex since $F(.\,,.)$ is convex. It is easy to see that problem (9.3) is a special case of problem (3.20). Hence, the Dantzig–Wolfe method can be applied here, with one infimal subproblem. To simplify the notation, we shall write $f(x) = \sum_{i=1}^{N} \phi_i(x_i)$ and $a_k = (a_{1k} \ldots a_{Nk})$. Suppose that S schedules (grid points) are available at some iteration t of the adjustment phase: $\hat{x}^1 \ldots \hat{x}^S$. The following supremal subproblem results:

$$\begin{aligned}
\text{Minimize} \quad & \sum_{s=1}^{S} f(\hat{x}^s)\lambda^s \\
\text{s.t.:} \quad & \sum_{s=1}^{S} (a_k \hat{x}^s)\lambda^s \leq U_k \qquad (k = 1 \ldots K), && (9.4a) \\
& -\sum_{s=1}^{S} (a_k \hat{x}^s)\lambda^s \leq -L_k \qquad (k = 1 \ldots K), && (9.4b) \\
& \sum_{s=1}^{S} \lambda^s = 1, && (9.4c) \\
& \text{all } \lambda^s \geq 0.
\end{aligned}$$

Let $\bar{\lambda} = (\bar{\lambda}^1 \ldots \bar{\lambda}^S)$ be an optimal solution to (9.4). Let π_k^1 and π_k^2 be simplex multipliers associated with (9.4*a*) and (9.4*b*). α is the simplex multiplier associated with (9.4*c*). The infimal subproblem then becomes:

$$\begin{aligned} &\text{Minimize} \quad f(x) - \sum_{k=1}^{K} (\pi_k^1 - \pi_k^2) a_k x \\ &\text{s.t.:} \qquad x \in X. \end{aligned} \tag{9.5}$$

The specific method employed to solve (9.5) will be commented on later. Assume one can find an optimal solution $\hat{x}^{S+1}$ to (9.5). If

$$f(\hat{x}^{S+1}) - \sum_{k=1}^{K} (\pi_k^1 - \pi_k^2) a_k \hat{x}^{S+1} - \alpha \geq 0, \tag{9.6}$$

an optimal solution to the original problem is already available, namely

$$\bar{x} = \sum_{s=1}^{S} \bar{\lambda}^s \hat{x}^s.$$

If (9.6) does not hold, the column associated with $\hat{x}^{S+1}$ is added to the supremal subproblem, and the stage is set for the next iteration of the adjustment phase.

Since this method usually does not converge in a finite number of steps, a termination criterion must be used. Let $\bar{z}$ denote the optimal solution value for the original problem, and z_t the optimal value of the supremal subproblem at iteration t. A lower bound for $\bar{z}$ is given by

$$\tilde{z}_t = z_t + f(\hat{x}^{S+1}) - \sum_{k=1}^{K} (\pi_k^1 - \pi_k^2) a_k \hat{x}^{S+1} - \alpha.$$

This lower bound is the same as the one given for the Dantzig–Wolfe method for LP problems (see section 3.3.5). If $\varepsilon > 0$ is some given tolerance level, then the following rule determines termination: Stop if $z_t - \tilde{z}_* < \varepsilon$, otherwise, continue ($\tilde{z}_*$ is the best lower bound obtained so far).

The infimal subproblem (9.5) is a nonlinear programming problem. It is hence, in its full generality, considerably more difficult to handle than the supremal subproblem. In the study of Chaly *et al.*, the power loss function took on a special form, enabling the application of an iterative process to find a solution to the Kuhn–Tucker conditions.

Chaly *et al.* state that this two-level method for power generation planning has been programmed and implemented on a BESM-4M computer (1974, p. 167).

In closing this section on the application of a two-level method to electrical power generation planning, some additional applications of multilevel methods to the same general problem area may be mentioned. Muckstadt and Koenig (1977) consider a power generation planning problem somewhat similar to the one discussed here, although more complex in its details. A

Lagrangean decomposition method is used. Cazalet (1970) discusses a problem relating to capacity expansion in a power plant system. A two-level method similar to Lagrangean decomposition is used. Noonan and Giglio (1977) discuss a problem relating to investments and power deliveries in a power system. The investment decisions are represented by 0–1 variables. Assuming fixed investment choices, an LP problem relating to power deliveries results. Benders decomposition is used to solve the overall problem.

REFERENCES

Appelgren, L. H. 1969. A column generation algorithm for a ship scheduling problem. Transportation Science 3: 53–68.

Appelgren, L. H. 1971. Integer programming methods for a vessel scheduling problem. Transportation Science 5: 64–78.

Cazalet, E. G. 1970. Decomposition of Complex Decision Problems with Applications to Electrical Power System Planning. Ph.D. dissertation. Stanford University.

Chaly, G. V., S. G. Zlotnik, A. I. Lazebnik, and G. V. Spiridonova. 1974. Algorithms for optimization of power systems schedules based on the simplex method. Kybernetes 3: 161–172.

Land, A. H., and A. G. Doig. 1960. An automatic method of solving discrete programming problems. Econometrica 28: 497–520.

Muckstadt, J. A., and S. A. Koenig. 1977. An application of Lagrangian relaxation to scheduling in power-generation systems. Operations Research 25: 387–403.

Noonan, F., and R. J. Giglio. 1977. Planning electric power generation: A nonlinear mixed integer model employing Benders decomposition. Management Science 23: 946–956.

10 Water Pollution Control

10.1 INTRODUCTION AND OVERVIEW

This chapter considers multilevel methods for an important problem in water systems management: pollution control This is but one of several problems in water systems management that have been tackled by multilevel methods. In fact, the literature on multilevel models and methods for water systems management is quite large, dating back at least to Dorfman (1962). Dorfman recognized rather soon after the invention of the Dantzig–Wolfe decomposition principle the potential usefulness of that method for problems regarding water systems. Haimes (1977) offers a survey of multilevel methods and models for water resources systems, and this chapter was designed so as to minimize overlap with Haimes's volume. Problems that are not discussed here, but are treated by Haimes, include capacity expansion in water resources systems (see also Nainis and Haimes 1975) and aquifer identification. Another problem that has been treated by multilevel methods is water utilization in a complex system consisting of rivers, reservoirs, and canals (Hall and Shephard 1967).

By way of introduction, imagine the following problem concerning a river or lake. There is a set of polluters (industrial installations, cities, and so on) and a Central Agency (CA) in charge of pollution control. Each polluter emits a certain quantity of polluted water per time unit. This effluent can be treated to a lesser or greater degree locally, by the polluter itself (in a local treatment plant). Certain minimal requirements have been formulated in advance for various water quality characteristics (e.g., dissolved oxygen). Suppose one wants to decide on treatment levels for each local polluter such that total treatment cost (summed over all treatment plants) is minimized.

This pollution control problem can be formulated as a mathematical programming problem. It turns out to be a straightforward resource-allocation problem: Allocate the natural assimilative capacity of the river or lake in such a

fashion that the required residual treatment is achieved at a minimal cost. The decision variables are the levels of treatment of the different polluters.

The pollution control problem can be solved in a two-level fashion, through an iterative information exchange between the CA and the polluters. As a result of such a two-level process, the CA can decide on a treatment level for each polluter, or, alternatively, a set of effluent charges that will induce each polluter to select a level of treatment that is desirable for the system as a whole. We note that this pollution control situation is very similar to the problem situation encountered in the discussion in Chapter 6 of two-level procedures for planning in divisionally organized corporations. In both cases, we have a resource-allocation situation, where the overall problem is one of finding optimal activity levels for the infimal decision units, and where those activity levels imply a particular allocation of certain jointly utilized resources.

From the point of view of multilevel systems analysis, the pollution control problem outlined here is particularly interesting in that a two-level institutional arrangement is suggested in the literature. That is, it is suggested that the problem could actually be solved through an iterative dialogue between the CA and the polluters. In other words, the multilevel procedure corresponds to a particular multilevel institutional arrangement. This, again, is reminiscent of multilevel procedures for business planning, where a definite multilevel institutional arrangement is also suggested, as was pointed out in Chapter 6.

It should be clear from the start that this chapter deals with multilevel aspects of one particular pollution control situation. Hence, an exhaustive treatment of the pollution issue is not attempted. In particular, the following question is not considered: How can the minimal requirements for various water quality characteristics (referred to above) be established?

In section 10.2, we present two multilevel approaches to the water pollution control problem of the Miami River (in Ohio, U.S.A.). The overall problem is formulated in section 10.2.1. A study of Hass (1970), who utilized the Dantzig–Wolfe method for nonlinear programs, is discussed in section 10.2.2. An alternative approach developed by Haimes *et al.* (1972), utilizing Lagrangean decomposition, is described in section 10.2.3.

10.2 THE MIAMI RIVER CASE

10.2.1 THE OVERALL PROBLEM

In this section, we discuss a pollution control problem taken from Hass (1970). A stretch of the Miami River (Ohio, U.S.A.) is considered.* The river is divided

* The Miami River case is also discussed in Haimes's volume (Haimes 1977, pp. 372–382). The emphasis in our discussion is a bit different from that of Haimes. See also Kulikowski (1973) and Mora–Camino (1977, pp. 88–94) for two other formulations of multilevel pollution control situations.

into 27 reaches, and there are 15 polluters altogether. The reaches are defined in such a way that each of them contains one polluter (but not two or more) or one tributary. Only one water quality characteristic is considered, level of dissolved oxygen (DO). It is required that the DO level be greater than or equal to 4 mg/liter in each reach. Let i index reaches ($i = 1 \ldots 27$) and j polluters ($j = 1 \ldots 15$). w_j is the BOD (biological oxygen demand) load introduced by the jth polluter. Note that the w_j are constants, i.e., not decision variables in the problem context considered here. Let a_{ij} denote the number of pounds of oxygen demanded in reach i to offset 1 pound of BOD discharged by polluter j. Naturally, $a_{ij} = 0$ if polluter j is located downstream from reach i. Let b_i denote the amount of DO available in reach i for the decomposition process (total available minus the minimal requirement of 4 mg/liter). Denote by x_j the percentage of w_j removed through treatment at source j. We can now formulate a restriction for each reach $i = 1 \ldots 27$:

$$a_{i1}w_1(1-x_1)+a_{i2}w_2(1-x_2)+\cdots+a_{i,15}w_{15}(1-x_{15}) \leq b_i. \tag{10.1}$$

What (10.1) says is that the demand for DO in each reach i must not exceed the supply. The demand depends on the decision variables $x_1 \ldots x_{15}$—that is, on the level of treatment by each polluter. Additionally, the following constraint is imposed for each x_j ($j = 1 \ldots 15$):

$$0.45 \leq x_j \leq 0.99. \tag{10.2}$$

The lower bound results from the requirement that each polluter undertake at least primary treatment (filtering, chlorination, and settling). Such treatment removes about 45 percent of the BOD load. The upper bound is a technical upper limit on the extent of purification possible.

The objective function is simply the sum of the treatment costs of the individual polluters:

$$\sum_{j=1}^{15} \phi_j(x_j), \tag{10.3}$$

where each $\phi_j(x_j)$ has been estimated as

$$\phi_j(x_j) = 160.8 + 26.7q_j + 640.7(x_j - 0.45)^2 + 255.7q_j(x_j - 0.45)^2. \tag{10.4}$$

The q_j are parameters denoting plant sizes and are given as constants (i.e., are not decision variables). The cost functions include capital costs (on an annual basis) and operating costs and are developed on the basis of engineering data. Equation (10.4) hence expresses costs associated with operating treatment plant j at intensity x_j. The total objective function (10.3) apparently expresses minimization of the total treatment cost along the river.

Assembling the objective function (10.3) and the restrictions (10.1) and (10.2), one obtains a nonlinear programming problem, the overall problem of

this section. The objective function is quadratic and convex. The constraints are linear. We note further that the total problem (10.1)–(10.3) is decomposable: the objective function is separable by index j, and each restriction (10.2) defines one subblock. The restrictions (10.1) are the coupling ones. If one assumes that the overall problem (10.1)–(10.3) has a feasible solution, then an optimal solution exists and is unique.

Data for the parameters a_{ij}, w_j, b_i, and q_j are provided in Hass's article (Hass, 1970). We will not concern ourselves here with how those data were derived but remark only that the data-gathering work is not trivial.

10.2.2 A PLANNING PROCEDURE BASED ON DANTZIG–WOLFE DECOMPOSITION

The problem (10.1)–(10.3) can be solved through, for example, the (nonlinear) Dantzig–Wolfe decomposition algorithm. This could be done in an institutional manner—that is, through an iterative information exchange between the CA and the 15 polluters, where each participating unit performs certain subproblem calculations at each iteration. That was *not* done in this particular case. Instead, Hass himself solved the problem by the nonlinear Dantzig–Wolfe method, in an attempt to simulate what the resulting information flows and final solution would have been if that method had been used as a planning tool by the CA and the polluters along the Miami River. That is, the purpose of Hass's study was exactly the same as that of the studies by Ljung and Selmer and by Christensen and Obel that were discussed in Chapter 6. It may be remarked here that, as far as is known, there are no reported implementations of planning procedures founded on decomposition algorithms for pollution control problems, just as there are no reported implementations for business planning in divisionalized corporations.

The nonlinear Dantzig–Wolfe method operates as follows in this case. Suppose n_j proposals have been obtained so far from polluter j ($j = 1 \ldots 15$). Let each such proposal be denoted (ϕ_j^s, x_j^s). The restricted master problem in the current iteration is then written as

$$\text{Minimize} \quad \sum_{j=1}^{15} \sum_{s=1}^{n_j} \phi_j^s \lambda_j^s$$

$$\text{s.t.:} \quad \sum_{j=1}^{15} \sum_{s=1}^{n_j} [a_{ij} w_j (1 - x_j^s)] \lambda_j^s \leq b_i \qquad (i = 1 \ldots 27), \tag{10.5}$$

$$\sum_{s=1}^{n_j} \lambda_j^s = 1 \qquad (j = 1 \ldots 15),$$

$$\lambda_j^s \geq 0 \qquad (j = 1 \ldots 15, \quad s = 1 \ldots n_j).$$

Let $\pi_i (i = 1 \ldots 27)$ be a dual multiplier associated with restriction (10.5), and set $p_i = \pi_i$. Then p_i may be interpreted as a tentative tax rate associated with polluting the ith reach in the next iteration. The infimal subproblems become (for $j = 1 \ldots 15$)

$$\begin{aligned} &\text{Minimize} \quad \phi_j(x_j) + w_j(1 - x_j)T_j \\ &\text{s.t.:} \qquad 0.45 \leq x_j \leq 0.99, \end{aligned} \tag{10.6}$$

where $T_j = \sum_{i=1}^{27} p_i a_{ij}$. T_j is hence a composite tentative tax rate facing polluter j in the next iteration. Note that the infimal subproblems (10.6) can be easily solved, since an optimum is found either at one of the boundary values (0.45 or 0.99) or at the unconstrained optimal value of the objective function. This unconstrained optimal value may be found by simple differentiation.

The restricted master problem does not converge finitely for a nonlinear problem. However, once a feasible restricted master problem has been obtained, the iterative process may be stopped, and a feasible solution to the original problem (10.1)–(10.3) may be recovered. If the Dantzig–Wolfe method is to be used as a planning tool in the current problem situation, then it is necessary that a "good" solution to the restricted master problem can be obtained in very few iterations. The reason is obviously that only a few iterations of information exchange would be undertaken in a real-world planning situation. This is the same requirement that was imposed in Chapter 6, in the discussion of the utilization of decomposition methods as the basis for planning procedures in divisionalized corporations.

Hass gives some information about the convergence performance of the Dantzig–Wolfe method in this case. To generate a feasible restricted master problem, the first two iterations used heuristic price vectors, the first of which was simply $p_i = 0$ for $i = 1 \ldots 27$. This resulted in a feasible restricted master problem in the third iteration. After four iterations, the value of the restricted master problem objective function was 8,616 (dollars/day), which should be compared with the true optimal solution value of the original problem (10.1)–(10.3) of 8,324. After six iterations, the restricted master problem objective function value was 8,317. It may hence be concluded that a good solution may, indeed, be obtained in a small number of iterations. In this particular case, it is partly due to the fact that the overall problem (10.1)–(10.3) is a small one and has a very simple structure.

If the information exchange between the CA and the polluters in the adjustment phase is halted after, for instance, four iterations, the question then arises of how the resulting decisions are to be implemented. (This question also arises in the business planning context; see section 6.2.4). That is, how is the execution phase to be carried out? In this case, the most natural way is perhaps for the CA to issue treatment levels to the polluters. That is, each polluter is informed that he must remove no less than a certain percentage of the BOD

discharged by him. Let those treatment levels be denoted by $\bar{x}_j$. Each polluter then solves the following infimal subproblem in the execution phase:

$$\begin{aligned} &\text{Minimize} && \phi_j(x_j) \\ &\text{s.t.:} && w_j(1-x_j) \leq w_j(1-\bar{x}_j), \\ & && 0.45 \leq x_j \leq 0.99, \end{aligned}$$

and implements the solution in actual treatment. This would correspond to a simplified version of the implementation form "right-hand-side allocations" discussed in section 6.2.4.

An alternative form of implementation would be through "tax rates" (corresponding to "prices" in the business planning context). Suppose $p_1^* \ldots p_{27}^*$ are optimal dual multipliers associated with the constraints (10.1). $p_1^* \ldots p_{27}^*$ cannot be obtained through the scheme discussed here, since the Dantzig–Wolfe method does not converge finitely and since it has been stated that only a small number of iterations can be undertaken in the present situation. However, for the sake of argument, suppose that $p_1^* \ldots p_{27}^*$ are announced to the polluters. Each polluter can then construct and solve the following infimal subproblem

$$\begin{aligned} &\text{Minimize} && \phi_j(x_j) + w_j(1-x_j)T_j^* \\ &\text{s.t.:} && 0.45 \leq x_j \leq 0.99, \end{aligned} \tag{10.7}$$

where $T_j^* = \sum_{i=1}^{27} p_i^* a_{ij}$. $[x_1(p_1^* \ldots p_{27}^*) \ldots x_{15}(p_1^* \ldots p_{27}^*)]$ is then the unique optimal solution to the original problem (10.1)–(10.3), where $x_j(p_1^* \ldots p_{27}^*)$ denotes the unique optimal solution to (10.7). The p_i^* define a set of tax rates with desirable properties: p_i^* measures the marginal damage to the total community (in terms of increased treatment cost) of dumping one additional unit of BOD into reach i. Each polluter, by solving (10.7), balances his marginal tax payment with his marginal treatment cost, and arrives at a decision that is optimal overall. We note in passing that these desirable properties of the tax rates would not hold for an overall problem of the linear type; this has been pointed out more than once in earlier chapters (see sections 2.1.2 and 6.2.4).

Suppose now that the iterative information exchange is halted after a limited number of iterations. In that case, the p_i^* will not be on hand, only a different, nonoptimal set of multipliers p_i', associated with the restrictions (10.5) of the restricted master problem in the last iteration. If the DW method converges rapidly, then one may hope that the p_i' are "close" to the p_i^*. In the present case, $p_{12}^* = 0.4132$, $p_{26}^* = 0.2118$, and $p_i^* = 0$ for all other indices i. $p_{12}' = 0.432$, $p_{26}' = 0.236$, and $p_i' = 0$ for all other indices i after six iterations. This means that the p_i' are actually quite close to the p_i^*. Now consider what happens if, in the execution phase, the tax rates p_i', obtained after six iterations, are announced to the polluters. That is, the polluters are instructed to

formulate their infimal subproblems (10.7), setting $p_i = p_i'$. These subproblems are then to be solved, and the solutions to be implemented in actual treatment levels. It turns out that the infimal subproblem solutions, $[x_1(p_1' \ldots p_{27}') \ldots x_{15}(p_1' \ldots p_{27}')]$, are very close to $[x_1(p_1^* \ldots p_{27}^*) \ldots x_{15}(p_1^* \ldots p_{27}^*)]$ (see Hass 1970, pp. 363–364). This is, in fact, what one would expect, since each $x_j(p_1 \ldots p_{27})$ is a continuous function of the p_i in this case [the objective function of (10.6) is *strictly* convex; see also section 3.7.1]. This means that the implementation form "tax rates" (or effluent charges) may be a reasonably good one in this case, even if the adjustment phase is terminated after a relatively small number of iterations of information exchange.

10.2.3 A LAGRANGEAN SOLUTION METHOD

In a paper by Haimes *et al.* (1972), the overall pollution control problem (10.1)–(10.3) was reconsidered, and different two-level method, a Lagrangean method, was proposed (section 3.7).

Let p_i^t $(i = 1 \ldots 27)$ be the tentative dual multipliers associated with restrictions (10.1) in iteration t of the adjustment phase. Given these multipliers, the following infimal subproblems [of the same form as (10.6) and (10.7)] are solved:

$$\begin{aligned} &\text{Minimize} && \phi_j(x_j) + w_j(1 - x_j)T_j^t \\ &\text{s.t.:} && 0.45 \le x_j \le 0.99, \end{aligned}$$

where $T_j^t = \sum_{i=1}^{27} p_i^t a_{ij}$. Let $x_j(p_1^t \ldots p_{27}^t)$ be the optimal solution.

The supremal subproblem then consists of adjusting the p_i^t. This adjustment was carried out in Haimes *et al.* (1972) in a manner slightly different from the procedure outlined in section 3.7.1. Let $L(x, p)$ be the Lagrangean function:

$$L(x, p) = \sum_{j=1}^{15} \phi_j(x_j) + \sum_{i=1}^{27} p_i \left\{ \sum_{j=1}^{15} a_{ij} w_j (1 - x_j) - b_i \right\}.$$

As in section 3.7.2, the dual function is defined as

$$h(p) = \min\{L(x, p) | 0.45 \le x_j \le 0.99, \qquad j = 1 \ldots 15\}.$$

The adjustment of the p_i^t, i.e., the calculation of new tentative multipliers p_i^{t+1}, is performed as follows. First, a direction of change is defined as (for $i = 1 \ldots 27$)

$$d_i^{t+1} = \max\left\{0; \left[\sum_{j=1}^{15} a_{ij} w_j (1 - x_j(p_1^t \ldots p_{27}^t)) - b_i\right]\right\} \qquad \text{if } p_i^t = 0;$$

$$d_i^{t+1} = \left[\sum_{j=1}^{15} a_{ij} w_j (1 - x_j(p_1^t \ldots p_{27}^t)) - b_i\right] \qquad \text{if } p_i^t > 0.$$

Next, a step size α^{t+1} is determined so as to maximize

$$h((p_1^t \ldots p_{27}^t) + \alpha^{t+1}(d_1^{t+1} \ldots d_{27}^{t+1}))$$

subject to the restrictions $\alpha^{t+1} \geq 0$ and $p_i^t + \alpha^{t+1} d_i^{t+1} \geq 0$ $(i = 1 \ldots 27)$. Let $\bar{\alpha}^{t+1}$ be the optimal step size. Then $p_i^{t+1} = p_i^t + \bar{\alpha}^{t+1} d_i^{t+1}$. The economic meaning of this adjustment is the same as in section 3.7.1: If the ith constraint (10.1) is violated, then the supply of natural assimilative capacity in reach i is smaller than the demand. In that case, the tax rate p_i^t should be increased. In the converse case, p_i^t is decreased. No tax rate is allowed to become negative, however.

In selecting the optimal step size $\bar{\alpha}^{t+1}$, a Fibonacci search procedure was used (Haimes *et al.* 1972, p. 766). This search procedure, as utilized by Haimes *et al.*, requires that the polluter cost functions $\phi_i(x_i)$ be known and at hand.

The purpose of this two-level price adjustment method is obviously to bring about the convergence of the p_i^t to the overall optimal dual multipliers p_i^* $(i = 1 \ldots 27)$. If the p_i^* can be obtained, then that is equivalent to solving the original problem, since $[x_1(p_1^* \ldots p_{27}^*) \ldots x_{15}(p_1^* \ldots p_{27}^*)]$ is an optimal solution to the original problem, as was pointed out in the preceeding subsection.

The computations in Haimes *et al.* (1972) were initiated with $p_i^1 = 0$ for $i = 1 \ldots 26$, and $p_{27}^1 = 5$. In the fourth iteration, $[x_1(p_1^4 \ldots p_{27}^4) \ldots x_{15}(p_1^4 \ldots p_{27}^4)]$ was already quite close to the optimal solution to the overall problem (Haimes *et al.* 1972, p. 767). This means that the iterative process could, in principle, have been halted at that point. This rapid convergence in the first iterations presumably depends to some extent on the optimal choice of α^t. The following iterations showed very slow convergence. The process was stopped after 99 iterations, at which point the p_i^t were very close to the p_i^*.

We note now that the computation of an optimal step size, α^t, which is part of the supremal subproblem in each iteration, requires that the polluter cost functions $\phi_i(x_i)$ be known, as already mentioned. If, in a real institutional setting, the polluters are unwilling, or unable, to specify these functions and send them to the CA, the above approach cannot be used as an *institutional* two-level method. The reason is, of course, that the CA will not have all the information at hand to solve the supremal subproblem. This is not so for the Dantzig–Wolfe method, where the CA does not need detailed knowledge of the functions $\phi_i(x_i)$ (as is evident from the discussion in the preceding subsection). If the CA does have detailed knowledge about the functions $\phi_i(x_i)$, then the rationale for using a two-level *institutional* problem-solving method is not so strong, but in such a case the approach of Haimes *et al.* can still be used by the CA as a *computational* aid (instead of a direct application of single-level nonlinear programming). The α^t can, of course, also be picked heuristically, in which case the CA need not know the $\phi_i(x_i)$. If so, a price adjustment, or Lagrangean, scheme can be used as an institutional two-level planning method.

However, convergence is probably slowed down, and it is not clear that a satisfactory solution to the original problem can be obtained in a small number of iterations.

10.3 CONCLUDING REMARKS

The discussion in this chapter has been rather similar to that in Chapter 6. In fact, the literature on institutional multilevel approaches to pollution control is quite similar to the literature on such approaches to planning in business corporations, even though two fairly distinct sets of authors are involved. For instance, in an article by Ferrar (1973), which contains a theoretical discussion of a multilevel approach to a pollution control problem of the type discussed in this section, the following issues are mentioned, all of which we recognize from the literature on multilevel business planning:

1. The distinction between adjustment phase and execution phase (Ferrar 1973, p. 174). That is, the various trial plans calculated by polluters and the CA in the adjustment phase are only steps on the way to the final and definitive one, implemented in the execution phase.
2. Cheating by infimal subunits (Ferrar 1973, p. 177).
3. The fact that polluters may be unwilling or unable to submit to the CA a complete description of their infimal subproblem specifications. This necessitates an institutional multilevel approach (i.e., makes it impossible for the CA to solve the overall pollution control problem directly, in a single-level fashion) (Ferrar 1973, p. 173).

In the overall problem (10.1)–(10.3), the objective function was taken as the sum of individual polluter treatment costs. This is obviously a rather peculiar objective function and points to one reason why it may be more difficult to implement multilevel planning procedures for pollution control than for business planning: If the polluters are separate organizations institutionally, then one could imagine that there would be considerable political problems in defining a suitable objective function. This difficulty does not arise to the same extent in the business planning situation, since a single organizational unit, a corporation, is involved. Hence, it is not unreasonable to define the total objective function as the sum of divisional contributions to profit. This indicates that the river pollution situation is considerably more complex from an institutional and political point of view than business planning, making the application of two-level methods—or any other methods, for that matter—more difficult. As a matter of fact, multilevel methods like the ones described here appear to have had little influence in practice. For instance, it appears that rather crude methods for pollution control and setting of effluent charges are

used for at least some European river basins (see the survey in Hoet-Mulquin 1974).

Some final remarks, relating to the use of effluent charges, should be made. In the literature on multilevel methods in water systems (see, e.g., Haimes 1973, p. 359; or Hass 1970, p. 355), it is often suggested that taxes on polluters may serve at least two different purposes. First, they have desirable incentive properties. This has already been discussed in the preceding subsections and, in fact, forms the basis for the two-level methods outlined there. A second purpose of effluent taxes is to raise revenue. That is, it is suggested that the taxes actually be paid to the CA, and then used by the CA to install additional treatment facilities (e.g., a dam for flow augmentation or a central treatment plant).

Taxes with desirable incentive properties may not always suffice to cover the costs of the proposed central treatment facility. For the case of the Miami River, Upton (1971) utilized the same data as Hass and Haimes *et al.* but allowed for the introduction of a flow augmentation reservoir. He then showed that taxes with optimal incentive properties would, indeed, not pay for the reservoir.

This issue is somewhat reminiscent of one treated in the accounting literature: the different purposes of transfer prices. Transfer prices may be used to motivate divisional managers to make good decisions, and also may be used to evaluate divisional performance. These are two frequently mentioned, *different* purposes of transfer prices. There seems to be some awareness among accounting theorists that *one* set of transfer prices cannot be made to serve *all* purposes simultaneously. Maybe the pollution control situation is analogous: Perhaps it is too much to expect one set of effluent charges to have desirable incentive and financial properties at the same time.

REFERENCES

Dorfman, R. 1962. Mathematical models: The multistructure approach, pp. 494–539. In A. Maas, M. M. Hufschmidt, R. Dorfman, H. A. Thomas Jr., S. A. Marglin, and G. M. Fair, Design of Water-Resource Systems. Cambridge, Massachusetts: Harvard University Press.

Ferrar, T. A. 1973. Nonlinear effluent charges. Management Science 20: 169–178.

Haimes, Y. Y. 1973. Decomposition and multilevel approach in the modeling and management of water resources systems, pp. 347–368. In D. M. Himmelblau (ed.), Decomposition of Large-Scale Problems. Amsterdam: North-Holland.

Haimes, Y. Y. 1977. Hierarchical Analyses of Water Resources Systems. New York: McGraw-Hill.

Haimes, Y. Y., J. Foley, and W. Yu. 1972. Computational results for water pollution taxation using multilevel approach. Water Resources Bulletin 8: 761–772.

Hall, W. A., and R. W. Shephard. 1967. Optimum Operations for Planning of a Complex Water Resources System. Contribution No. 122. Water Resources Center, University of California at Los Angeles.

Hass, J. E. 1970. Optimal taxing for the abatement of water pollution. Water Resources Research 6: 353–365.

Hoet-Mulquin, M. E. 1974. Les Redevances d'Effluent: Instrument d'une Politique Anti-Pollution Coherente. Document de Travail CB 1/10–CB 3/5, CORE. Louvain: Université Catholique de Louvain.

Kulikowski. R. 1973. Optimization of the decentralized large-scale pollution control model. Bulletin de l'Academie Polonaise des Sciences, Sciences Techniques, 21: 39–45.

Mora-Camino, F. 1977. Contribution à l'Analyse et à la Commande des Systèmes Socio-Economiques. Ph.D. dissertation. Université Paul Sabatier, Toulouse.

Nainis, W. S., and Y. Y. Haimes. 1975. A multilevel approach to planning for capacity expansion in water resource systems. IEEE Transactions on Systems, Man, and Cybernetics 5: 53–63.

Upton, C. 1971. Application of user charges to water quality management. Water Resources Research 7: 264–272.

11 Conclusion

11.1 PROBLEM STRUCTURES AND SOLUTION METHODS

In Chapters 5–10, a number of actual cases in which multilevel methods have been used for solving problems in economics and management were presented. In section 1.4, we posed certain requirements that a case study should satisfy in order to qualify for inclusion (e.g., real-world data should be involved). It is not easy to find good applications of multilevel methods to real-world problems that satisfy these requirements. Nevertheless, some noteworthy examples of applications have been left out and should therefore be mentioned briefly here.

One apparently successful implementation of column generation is to cutting-stock problems (Gilmore and Gomory 1961, 1963). This application is quite well known among management scientists, and for this reason we have not included it in this volume. In the management of power systems and water systems there are also applications of multilevel methods, as already indicated at the end of Chapter 9 and the beginning of Chapter 10. Additionally, there are applications of multilevel methods in world modeling (Mesarovic and Pestel 1974a, b; for a critical review of multilevel world modeling see also Fedanzo 1976).

One further study that deserves mention is that of Manheim (1966), who developed a hierarchical method for locating highways. The rationale of this method is similar to that underlying hierarchical production planning (discussed in Chapter 7)—a disaggregation scheme with each level representing a certain degree of disaggregation.

In this chapter, we will try to answer the question: How useful are multilevel methods for solving problems in economics and management? First, however, we will summarize some of the discussion in preceding chapters by identifying typical problem structures and multilevel solution methods that have been encountered.

Three typical problem structures can be mentioned:

1. LP problems with many columns. Or, more exactly, problems that, after suitable simplifications, result in LP formulations with many columns. Such problems were treated in section 7.2 (production planning), section 8.3 (planning of production and distribution), and section 9.2 (ship scheduling). In this situation, column generation is an obvious two-level method that can be used. The infimal subproblems are often of the type: Find the shortest (or longest) route through a network. The infimal subproblems were of this type in the production and distribution planning problem in section 8.3 and the ship scheduling problem in section 9.2.
2. Block-angular LP problems, and nonlinear generalizations of such problems. This is the "classical" problem structure for applying decomposition methods. It arises very naturally, for instance, in connection with national economic planning and planning in business corporations, as seen in Chapters 5 and 6. The coupling constraints impose conditions on all subunits (sectors of the economy, or divisions in a corporation) taken together. In addition, each sector or division is constrained by some local conditions. Obvious candidate methods for solving such problems are Dantzig–Wolfe decomposition and Lagrangean decomposition.
3. Mixed-integer programs. Often, the integer variables are of 0–1 type and represent investments or capacity acquisitions. The "linear" variables then represent operating decisions. One special subcase of this structure involves linear variables representing transportation activities. That is, if the investment variables are fixed, one obtains a set of independent transportation problems. This structure has been encountered only once in this volume, in section 8.2 (distribution system design). One may, however, find several additional examples of this investment-transportation structure, for instance in the Soviet literature (see, e.g., Zavel'skii *et al.* 1974). For mixed-integer problems (with and without the transportation feature), Benders decomposition may be used.

The typical multilevel solution methods mentioned here—i.e., column generation (often with infimal subproblems of the shortest-route type), Dantzig–Wolfe decomposition (linear and nonlinear), and Benders decomposition—were outlined in Chapter 3, and their application was illustrated in Chapters 5–10. Chapters 5–10 probably do not present an entirely fair picture of the relative usefulness of the different methods in one respect, though: Lagrangean decomposition was used in only one example study (section 10.2, on river pollution). This is not quite representative of the importance of that method. For certain classes of control-theoretic problems, the Lagrangean method may be the most promising one, as was mentioned in section 3.9.2.

In addition, multilevel methods that must be regarded as heuristic have been encountered, in particular in Chapter 5, on national and regional economic

planning. Hierarchical production planning, discussed in Chapter 7, may also be considered a heuristic method.

It should be stressed that the multilevel methods described in this volume are conceptually fairly simple ones. That is, they may largely be regarded as extensions of linear programming (one exception is Lagrangean decomposition). The level of technical–mathematical sophistication involved is moderate—certainly lower, for example, than that required by control theory or stochastic inventory theory. Moreover, the methods used in this volume are not unrelated. There are, on the contrary, connections between several of them: Dantzig–Wolfe decomposition is an extension of column generation (in that column generation is applied to the equivalent extremal problem). Benders decomposition may, in a certain sense, be regarded as dual to Dantzig–Wolfe decomposition, as demonstrated in section 3.5.5. Kornai–Liptak decomposition may be viewed as a simplified version of the Benders method. Again, though, Lagrangean decomposition stands a little apart from the other methods. The application studies in Chapters 5–10 represent some fairly important problem situations in economics and management. Thus, even though the methods treated in this book are fairly simple ones, and partially founded on a limited number of common ideas, they enable the analyst to reach quite far, as evidenced by the range of applications exhibited here.

In two respects, though, the conceptual simplicity of the methods may be a bit misleading. That is, we have glossed over certain technicalities. In the first place, as already pointed out in section 1.4, we have often not been very specific about how to solve the individual subproblems when using the various methods. For instance, when using the nonlinear Dantzig–Wolfe method, the infimal subproblems could be difficult nonlinear problems. Second, implementing these multilevel methods on a computer is usually not a trivial task; in most cases, it requires fairly skilful computer programming, at least if an efficient implementation is to be achieved. Again, we have not had much to say about implementation tactics.

Still another difficulty in applying multilevel methods concerns behavioural aspects of implementation. That is, applying some multilevel method in a real-world organization poses some difficulties for the systems analyst. At the very least, it demands certain interpersonal skills, so that he can communicate effectively with the potential users of the method in the organization. This, too, is a matter that we have glossed over. This issue—behavioural aspects of implementation—is by no means unique to multilevel methods; it is important for operations research and systems analysis methods in general.

It may be noted that all problem formulations in this volume have been deterministic ones. This should not be taken as evidence that the methods involved are unable to handle stochastic problems. The multilevel methods discussed here can, in fact, sometimes be extended to handle stochastic problem formulations (see, for instance, Jennergren 1973, where the Dantzig–

Wolfe method is generalized to solve a class of stochastic resource-allocation problems). It is simply that deterministic modeling, rather than stochastic, is convenient and useful for the planning problems considered here.

11.2 AN EVALUATION OF THE USEFULNESS OF MULTILEVEL METHODS

In attempting to evaluate the usefulness of multilevel methods, one can apply different criteria (or evaluation methods). First, one can compare multilevel and single-level methods, as applied to the same overall problems. Unfortunately, there are not too many such comparisons published in the literature. Chapter 4 reviewed some comparisons of Dantzig–Wolfe decomposition and ordinary LP applied to various test problems. Based on these comparisons, Dantzig–Wolfe decomposition appears to be of questionable usefulness. (This may not be so for *nonlinear* Dantzig–Wolfe decomposition, since for nonlinear problems there is usually no powerful and obvious single-level competitor, like ordinary LP in the linear case.) Some similar comparisons between column generation and ordinary LP have also been made, and were mentioned in sections 8.3.3 and 9.2.3. In the two cases discussed here, column generation outperformed single-level LP. This indicates that multilevel methods are useful at least in certain situations.

A second mode of evaluating multilevel methods is to look at application examples. In the application cases of this volume, there are usually no comparisons made between multilevel and single-level methods applied to the same overall problem, as already indicated. Nevertheless, one can try to evaluate the cases themselves in an "absolute" sense: Are they convincing case studies? Do they represent successful implementations of systems analysis in general?

A systems analysis study may be considered successful, if the results are adopted by the decision makers involved. In a recent discussion of the implementation of operations research methods, Huysmans (1975) identifies three degrees of adoption that are relevant for the cases presented here. In accordance with Huysmans' classification, we have the following three degrees of success in implementing multilevel methods, based on the management response to the solution proposals ("management" is to be interpreted in a broad sense):

1. The proposals result in "management action." That is, the results are accepted and implemented.
2. The proposals lead to "management change." That is, the solution is not only implemented, but the inherent logic of the multilevel model system becomes an integral part of management's thinking.

3. The implementation is so successful that it induces further applications of multilevel methods within the organization.

In other words, our argument is the following: If the application cases presented earlier are successful by these criteria (especially as evidenced by the two highest degrees of adoption), then that is to a certain extent due to the multilevel methods themselves. If there are several instances of successful implementation of multilevel methods, one may infer that those methods are useful. We will therefore examine the case studies of this volume in the light of this classification.

It is easy to identify those studies that do not meet the first degree of adoption. The planning studies of Chapter 6 are of this kind. They were intended to shed light on certain issues concerning the use of decomposition techniques as the basis for organizational planning procedures. The multilevel approaches to pollution control (Chapter 10) also belong to this group. In these studies, there is a correspondence between the subproblem hierarchy and the organizational hierarchy, and this correspondence is utilized in the decision-making process. The relevance of multilevel methods in this setting will be commented on later.

However, in several of the studies, implementation of the first degree ("management action") did apparently occur. From our interpretation of the sources, this occurred in the application of column generation to production planning (section 7.2), in both cases of distribution system planning (Chapter 8), and in the freight ship scheduling problem of section 9.2. The extent to which hierarchical production planning (section 7.3) has been implemented, leading to management action, is not clear, since only trial runs have been reported. With respect to the cases in national and regional economic planning in Chapter 5, one must be a bit careful about the meaning of "management action." One can hardly expect policymakers to rely on quantitative models alone in planning an economy or significant parts thereof. Therefore, management action should be considered to have occurred in such a situation if the results were discussed with policymakers and hence may have had an effect in shaping the actual decisions. The study of multilevel planning of the Mexican economy (section 5.3) illustrates this kind of implementation. The results of the "man–machine planning" study in Hungary (section 5.2) were also discussed with policymakers, but the actual impact of these discussions is uncertain. Whether the study of regional planning in section 5.4 has been implemented cannot be assessed from the documentation. In light of all this, it would seem fair to conclude that a good number of the cases discussed here have been implemented in the sense of leading to management action.

Whether the multilevel methods involved were adopted to such an extent as to result in "management change" is more difficult to say. To illustrate this point, consider the study of the determination of optimal

production–distribution programs (section 8.3). The logic of the two-level model system is based on the idea of viewing the problem as an integrated one. The relevant decision variables are the production quantities of individual commodities, identified by production origin, demand destination, and transport route. The potential of the analyses that can be carried out based on the model system will be realized only if management is willing to accept this integrated approach as part of its own thinking. Whether this actually happened is not reported. The freight ship route scheduling problem is similar in this respect, although we surmise that the use of a multilevel approach in this case did lead to management change, since an operating system with the possibility of manual intervention was set up by the company in question (see section 9.2.4). Unfortunately, the case studies are not explicit on this issue.

There is also unclear evidence about the third degree of adoption (repeated applications of multilevel methods within the same organization). However, after their first investigation (based on the Kornai–Liptak algorithm), Kornai and his associates continued their work on multilevel methods for national economic planning (section 5.2). This indicates that at least in this situation, multilevel methods were judged interesting enough for continued use. Also, the column generation method for production scheduling (section 7.2) was implemented in several factories of one company, indicating repeated applications (Ladson 1974, p. 41).

All in all, the reader may be disappointed by the relatively sparse evidence of successful real-life implementations of multilevel systems analysis. Scarcity of evidence, though, is a recurring phenomenon, as regards the practical use of quantitative planning methods. If this is kept in mind, multilevel systems analysis does not score too badly, since there is evidence of several successful implementations in the sense of "management action," and in a few of the cases, there is also some indication of successful implementation as measured by higher degrees of adoption.

A third way of evaluating the usefulness of multilevel systems analysis is to examine some arguments commonly made in favor of multilevel methods. That is, certain advantages of multilevel methods have been suggested in the literature, and one may now inquire if these advantages hold up in the light of the application cases.

Haimes (1977, pp. 60–63) lists several attributes of multilevel methods, which he claims reveal advantages over more conventional, single-level ones. We mention some of the more important attributes identified by Haimes and comment on their significance in view of the materials covered in this volume.

1. Conceptual simplification of complex systems. When a complex overall problem is decomposed into a subproblem hierarchy, a conceptual simplification may be achieved. This is particularly true when one analyzes highly coupled systems (i.e., systems with significant interactions between

various subunits). This attribute of multilevel methods is more pronounced for technically oriented overall problems (see section 3.9.1, on static multilevel control problems), but it does not seem to be of overriding consequence for problems in economics and management. The subproblem interactions in our case studies are, in general, fairly straightforward.

2. Reduction in dimensionality. This is clearly an important attribute of multilevel methods. Reduction in dimensionality does not always entail computational superiority, though. This is brought out in the discussion in Chapter 4 of the relative performance of the Dantzig–Wolfe method and ordinary LP. Nevertheless, multilevel methods are sometimes resorted to out of sheer necessity for dimensionality reduction. Recall, for instance, from the discussion in section 3.6 that the Kornai–Liptak method was designed precisely for this reason. Similarly, the Benders algorithm was used to solve the mixed-integer programming problem formulated in section 8.2 because that problem could not be handled in a single-level fashion by existing mixed-integer programming codes. Hence, multilevel methods are often appropriate for solving large-scale problems, although powerful single-level methods (e.g., compact inverse LP methods) sometimes constitute a viable alternative. This advantage, dimensionality reduction, could become less important as more powerful single-level methods (e.g., more powerful mixed-integer programming methods) are developed. The Hungarian experience points to this; multilevel methods were abandoned as more powerful LP codes became available (section 5.2.3).

3. More realistic modeling. Sometimes, complex problem situations involve nonlinear components that would have to be linearized in order to allow the application of a single-level technique, such as ordinary LP. This loss in accuracy can be avoided by the use of multilevel methods. One illustration of this is provided by the production planning problem described in section 7.2. In that problem, the nonlinear infimal subproblems could be solved by dynamic programming, whereas a single-level solution procedure would probably have been feasible only after some suitable linearization. But linearization means that the set-up costs would have to be neglected. This advantage of multilevel methods should be interpreted with care, though. Nonlinear subproblems (like those that arise in the nonlinear Dantzig–Wolfe method) are often difficult to solve, and they may have to be solved many times. In certain situations, this attractive attribute may be of only theoretical value.

4. The possibility of exploiting special problem structures. In this volume, we have seen several examples of how specific problem structures can be exploited through the use of multilevel systems analysis. For example, we have seen how infimal subproblems of the transportation type could be identified (section 8.2), and shortest-path problems have also been encountered (sections 8.3 and 9.2). These examples illustrate how, through a proper decomposition into a subproblem hierarchy, structures are revealed that can be exploited by

special solution techniques. This attribute probably largely accounts for some of the most successful implementations of multilevel systems analysis.

5. Flexibility. This advantage is rather obvious, and partially follows from the preceding one, the possibility of exploiting specific problem structures. Once a subproblem hierarchy has been formulated, each subproblem can be solved by the most appropriate method, irrespective of the overall problem formulation: Various techniques of network optimization, dynamic programming, and mathematical programming can be combined to handle the given overall problem.

So far in this section, we have examined the value of multilevel systems analysis in solving complex decision problems in economics and management. We have been concerned with multilevel methods mainly as *computational* tools. Comparisons of multilevel and single-level methods applied to the same problems, the success of selected case studies, and certain important attributes of multilevel methods lead us to conclude that multilevel systems analysis is indeed a useful methodology in certain problem situations. It is not a universally useful methodology, but the range of situations in which it can be applied is indicated by the case studies in Chapters 5–10.

In addition to the value of multilevel methods as computational tools, there are two further reasons for interest in multilevel systems analysis. First, it may be valuable for *modeling* only, as opposed to problem solving. That is, in certain situations one is interested only in constructing an adequate description, not in solving some overall decision problem. It may then be useful to construct a multilevel model—a representation in terms of a subproblem hierarchy, with associated information flows. This manner of modeling forces one to identify subcomponents and their interrelationships in the real-world situation, perhaps leading to valuable insights into the system structure. This possibility is exemplified by some studies (Malone 1972, Richardson and Pelsoci 1972), although not always entirely convincingly. Baumgartner *et al.* (1976) make an emphatic statement of the virtues of multilevel modeling in the social sciences.

A second additional reason for interest in multilevel systems analysis derives from institutional and economic interpretations and analogies. It has been mentioned several times in this volume that certain multilevel methods, such as the Dantzig–Wolfe decomposition principle, can be interpreted as a kind of formalized budgeting procedure. One could hence construct institutional decision-making procedures founded on these methods, to be implemented through an actual iterative dialogue between different subunits in the organization. The discussion in Chapters 6 and 10 was intended to shed light on the consequences of using such decision-making procedures in two situations: production and sales planning in business corporations, and river pollution control. The case studies involved were not real implementations but "simula-

tions." Whether decision-making procedures of this type will ever become important for real organizations it is too early to say. The results in Chapter 6 were not altogether encouraging. The river pollution case study in Chapter 10 could also be criticized for a certain naiveté in the assumptions about the underlying political–organizational situation (section 10.3). We are skeptical about the usefulness of institutional planning procedures founded on multilevel methods. However, rather than taking a definite stand on this issue, we wish to point out only that there is a danger in pushing the institutional interpretations of multilevel systems analysis methods too far. It may lead, for example, to inferences about organizational design that are simply not warranted.

11.3 A FINAL WORD

There is yet another argument in favor of multilevel systems analysis. The methods discussed in this volume are conceptually quite simple, as already mentioned (section 11.1). Related to this simplicity is a certain elegance. There is no question that, for instance, the column generation method for maximal multicommodity network flow problems of Ford and Fulkerson (discussed in section 3.2) appeals strongly to the aesthetic sense of the quantitatively oriented systems analyst. The same is true for the Dantzig–Wolfe and Kornai–Liptak decomposition methods. Actually, there is no doubt that systems analysts, operations researchers, and mathematical economists have been fascinated by the elegance of these decomposition methods, a fact also reflected in the literature. In contrast, factorization methods for linear programming, which are sometimes a powerful alternative to multilevel methods, are messy and rather inelegant.

We suggest that this elegance may be another good reason for becoming acquainted with multilevel systems analysis.

REFERENCES

Baumgartner, T., T. R. Burns, L. D. Meeker, and B. Wild. 1976. Open systems and multi-level processes: Implications for social research. International Journal of General Systems 3: 25–42.

Fedanzo, A. J., Jr. 1976. Multilevel, hierarchical world modeling. Technological Forecasting and Social Change 9: 35–49.

Gilmore, P. C., and R. E. Gomory. 1961. A linear programming approach to the cutting-stock problem. Operations Research 9: 849–859.

Gilmore, P. C., and R. E. Gomory. 1963. A linear programming approach to the cutting-stock problem—Part II. Operations Research 11: 863–888.

Haimes, Y. Y. 1977. Hierarchical Analyses of Water Resources Systems. New York: McGraw-Hill.

Huysmans, J. 1975. Operations research implementation and the practice of management, pp. 273–289. In R. L. Schultz and D. P. Slevin (ed.), Implementing Operations Research/Management Science. New York: American Elsevier.

Jennergren, L. P. 1973. A note on a Dantzig–Wolfe decomposition-like method for solving a particular resource-allocation problem under uncertainty. Mathematische Operationsforschung und Statistik 4: 127–132.

Lasdon, L. S. 1974. Generalized upper bounding methods in production scheduling and distribution, pp. 25–41. In R. Cottle and J. Krarup (ed.), Optimization Methods for Resource Allocation. London: The English Universities Press.

Malone, D. W. 1972. Modeling air pollution control as a large scale, complex system. Socio-Economic Planning Sciences 6: 69–85.

Manheim, M. L. 1966. Hierarchical Structure: A Model of Design and Planning Processes. Cambridge, Massachusetts: MIT Press.

Mesarovic, M., and E. Pestel. 1974a. Mankind at the Turning Point. New York: Dutton.

Mesarovic, M., and E. Pestel. 1974b. Multilevel Computer Model of World Development System, SP-74-1–SP-74-6. Laxenburg, Austria: International Institute for Applied Systems Analysis.

Richardson, J., and T. Pelsoci. 1972. A multilevel approach and the city: A proposed strategy for research, pp. 97–131. In M. D. Mesarovic and A. Reisman (ed.), Systems Approach and the City. Amsterdam: North-Holland.

Zavel'skii, M. G., M. R. Mazin, and L. E. Pochinshchikov. 1974. Experience in solving optimization problems of large size. (In Russian.) Ekonomika i matematicheskie metody 10: 285–295.

Index

Adjustment phase 17–25, 53, 134, 137, 142, 153, 198, 202
Aggregate production planning 156
Aggregation (disaggregation) 23–24, 99, 122, 156, 162–163, 169–170, 205
Arc-chain formulation 32–33, 36

Benders algorithm 22, 24, 28, 56–70, 78, 84, 120, 124, 128, 135, 149, 174–178, 193, 206–207, 211
Block-angular structure 12, 48, 54, 66, 84–86, 90, 101, 109, 112, 117, 123, 132, 138, 206
Block-product algorithm 85
Bound on solution value 46–48, 50–52, 56, 61, 71–72, 87–89, 142, 192
Branch-and-bound method 189

Chain 32
Cheating 23, 153
Column generation 5–6, 20, 24, 27–36, 39, 41, 45, 157–162, 172, 180–181, 183, 185, 187–189, 205–208, 213
Compact inverse methods 27, 36–37, 211
Compositional approach 20
Control theory 8, 28, 76–81, 206
Convex programming 166, 168, 183, 191
Coordinability 12–21, 74–75, 143, 161
Corporate constraints 133, 138–141, 143–144, 146–147, 152
Coupling constraints 12, 48, 74, 77, 79–80, 87, 93, 95, 101, 108–109, 114, 118, 133, 197, 206
Cutting-stock problem 205

Dantzig–Wolfe method for linear programs 5–6, 17–18, 24, 27, 32, 36–54, 56–57, 59, 61, 69–70, 84–97, 100, 104, 108–112, 122, 124, 134–136, 140–145, 147–150, 152–153, 192, 194, 206–208, 212–213
Dantzig–Wolfe method for nonlinear programs 27, 54–56, 183, 190–192, 195, 197–201, 206, 208, 211
Decentralization 23–24
Decomposition 5, 23–25, 28, 84
Decompositional approach 20
Direct decomposition 85, 86, 92
Distribution system 22, 172–181, 206, 209
Dominant schedules 158–161
Dual coordination method 77–78
Dual function 75, 200
Dynamic programming 7, 30, 127, 160, 187–188, 212
Dynamic systems 78

Economic systems debate 4–5
Effluent charges 195, 200, 203
Electricity generation 7, 183, 190–193

Everett's theorem 130
Execution phase 17–21, 24–25, 53, 91, 142, 198–199, 202
Extremal problem 41

Factorization methods 84–85, 93, 213
Farkas's lemma 57
Farm irrigation problem 8
Ford–Fulkerson method for maximal multicommodity network flow problems 5, 27, 31–32, 45, 172, 213
Freight ship route scheduling 7, 21, 30–31, 183–189, 206, 209–210
Full master problem 41, 59, 63, 151

Game theory 23, 72
Generalized GUB 85, 93
Generalized upper bounding (GUB) 85, 87, 162, 178, 181
Grid linearization 54

Heuristic multilevel methods 25, 28, 76, 100, 108, 110–111, 124, 128, 130, 206–207
Hierarchical production planning 21, 24, 158, 162–171, 205, 207, 209
Hierarchical systems theory 4, 6

Infimal subproblem 11–21
Integer programming (also mixed) 28, 56–57, 63, 69, 113–114, 120, 124, 159, 173–174, 176, 184–185, 188–189, 206, 211
Iteration 30

ten Kate method 68, 135–136, 146, 149–153
Knapsack problem 166
Kornai–Liptak algorithm 28, 70–72, 104–105, 108, 112, 135, 207, 210–211, 213
Kuhn–Tucker conditions 167, 192

Lagrangean decomposition 25, 28, 72–76, 78–79, 135, 193, 195, 200–201, 206–207
Lagrangean function 74–75, 200
Longest-path problem 187, 206

Man-machine planning 97, 108, 110, 124, 209
Maximal multicommodity network flow problem 5, 22, 24, 27, 31–36, 45, 180, 185, 213
Maximal single-commodity network flow problem 31
Minimal-cost multicommodity network flow problem 31, 172, 180–181
Multiproduct lot-size scheduling problem 157
Multistage optimization 100

National economic planning 6, 20–21, 99–125, 137, 206, 209
Node-arc formulation 32–33, 36, 180, 181
Nonlinear programming 7, 54–55, 73, 192

Ohse dual algorithm 93
On-line control 78, 81
Open-loop control 78
Organizational design 4, 23, 154, 213
Original problem 10–21
Overall problem 10–21

Paper board factory planning problem 6, 137–145
Parametric method 77–78
Partitioning methods 27
Penalty function method 77–78
PERT network 125–126
Plant-location model 173
Polyhedral convex cone 37–38, 43, 57, 67
Polyhedral convex set
Price-adjustment procedure 74–76, 200–201
Price-directive planning procedures 135
Primal decomposition methods 78
Production-distribution planning problem 172, 179–181, 206

Regional planning 6, 100, 125–130, 209
Resource-directive planning procedures 135
Restricted master problem 41, 55
Rolling horizon 164
Rosen algorithm 93

Shortest-path problem (algorithm) 5, 30, 34–36, 180–181, 206, 211
Single-product lot-size scheduling problem 160
Slater constraint qualification 73
Slaughterhouse planning problem 6, 145–153
Steepest-ascent algorithm 76
Suboptimization 114, 120, 125
Subproblem hierarchy 2, 11–18, 20–22, 69, 74, 140, 143, 147
Supremal subproblem 11–21

Three-level subproblem hierarchy 21, 162–163, 168–169, 189
Transfer price problem 16–17
Transportation problem 43–46, 84, 175, 206, 211

Water pollution control 7, 22, 153, 194–203, 213
World modeling 205

THE WILEY IIASA INTERNATIONAL SERIES ON APPLIED SYSTEMS ANALYSIS

Conflicting Objectives in Decisions
Edited by David E. Bell, *University of Cambridge;* Ralph L. Keeney, *Woodward-Clyde Consultants, San Francisco; and* Howard Raiffa, *Harvard University*

Material Accountability: Theory, Verification, and Applications
Rudolf Avenhaus, *Nuclear Research Center Karlsruhe and University of Mannheim*

Adaptive Environmental Assessment and Management
Edited by C.S. Holling, *Institute of Animal Resource Ecology, University of British Columbia*

Organization for Forecasting and Planning: Experience in the Soviet Union and the United States
Edited by William R. Dill, *New York University; and* G. Kh. Popov, *Moscow State University*

Management of Energy/Environment Systems: Methods and Case Studies
Edited by Wesley K. Foell, *University of Wisconsin-Madison*

Systems Analysis by Multilevel Methods: With Applications to Economics and Management
Yvo M.I. Dirickx, *Twente University of Technology; and* L. Peter Jennergren, *Odense University*

Connectivity, Complexity, and Catastrophe in Large-Scale Systems
John L. Casti, *New York University*

Forthcoming

Pitfalls of Analysis
Edited by G. Majone *(Italy); and* E.S. Quade *(USA)*

Control and Coordination in Hierarchical Systems
W. Findeisen *(Poland);* F.N. Bailey *(USA);* M. Brdyś, K. Malinowski, P. Tatjewski, *and* A. Woźniak *(all of Poland)*

Computerized Urban Traffic Guidance and Control Systems
Edited by H. Strobel *(GDR)*

Ecological Policy Design: A Case Study of Forests, Insects, and Managers
C.S. Holling, G.L. Baskerville, W.C. Clark, D.D. Jones, *and* C.A. Miller *(all of Canada)*

JOHN WILEY & SONS
Chichester · New York · Brisbane · Toronto
A Wiley–Interscience Publication

ISBN 0 471 27626 X