EQUITABLE RESOURCE ALLOCATION
The Information and Communication Technology (ICT) book series focuses on creating useful connections between advanced communication theories, practical designs, and end-user applications in various next generation networks and broadband access systems, including fiber, cable, satellite, and wireless. The ICT book series examines the difficulties of applying various advanced communication technologies to practical systems such as WiFi, WiMax, B3G, etc., and considers how technologies are designed in conjunction with standards, theories, and applications.

The ICT book series also addresses application-oriented topics such as service management and creation and end-user devices, as well as the coupling between end devices and infrastructure.

**T. Russell Hsing, PhD,** is the Executive Director of Emerging Technologies and Services Research at Telcordia Technologies. He manages and leads the applied research and development of information and wireless sensor networking solutions for numerous applications and systems. Email: thsing@telcordia.com

**Vincent K.N. Lau, PhD,** is Associate Professor in the Department of Electrical Engineering at the Hong Kong University of Science and Technology. His current research interest is on delay-sensitive cross-layer optimization with imperfect system state information. Email: eeknlau@ee.ust.hk

*Wireless Internet and Mobile Computing: Interoperability and Performance*
Yu-Kwong Ricky Kwok and Vincent K. N. Lau

*Digital Signal Processing Techniques and Applications in Radar Image Processing*
Bu-Chin Wang

*The Fabric of Mobile Services: Software Paradigms and Business Demands*
Shoshana Loeb, Benjamin Falchuk, and Euthimios Panagos

*Fundamentals of Wireless Communications Engineering Technologies*
K. Daniel Wong

*RF Circuit Design, Second Edition*
Richard Chi-Hsi Li

*Networks and Services: Carrier Ethernet, PBT, MPLS-TP, and VPLS*
Mehmet Toy

*Equitable Resource Allocation: Models, Algorithms, and Applications*
Hanan Luss
EQUITABLE RESOURCE ALLOCATION

Models, Algorithms, and Applications

Hanan Luss
AT&T Labs and Telcordia Technologies (retired)
To Dalia
My best friend for five decades,
but who is counting?
## CONTENTS

**Preface** ........................................ xi

**Acknowledgments** ................................ xvii

### 1 Introduction ................................. 1

1.1 Perspective / 1

1.2 Equitable Resource Allocation: Lexicographic Minimax (Maximin) Optimization / 3

1.3 Examples and Applications / 14

1.3.1 Allocation of High-Tech Components / 14

1.3.2 Throughput in Communication and Computer Networks / 15

1.3.3 Point-to-Point Throughput Estimation in Networks / 18

1.3.4 Bandwidth Allocation for Content Distribution / 20

1.3.5 Location of Emergency Facilities / 23

1.3.6 Other Applications / 25

1.4 Related Fairness Criteria / 26

1.5 Outline of the Book / 30

1.5.1 Chapter 2: Nonlinear Resource Allocation / 30

1.5.2 Chapter 3: Equitable Resource Allocation: Lexicographic Minimax and Maximin Optimization / 30

1.5.3 Chapter 4: Equitable Resource Allocation with Substitutable Resources / 31

1.5.4 Chapter 5: Multiperiod Equitable Resource Allocation / 32

1.5.5 Chapter 6: Equitable Network Resource Allocation / 33

1.5.6 Chapter 7: Equitable Resource Allocation with Integer Decisions / 34

1.6 Concluding Remarks and Literature Review / 35

1.6.1 Equitable Allocation of High-Tech Components / 38

1.6.2 Equitable Throughput in Communication and Computer Networks / 38
1.6.3 Point-to-Point Throughput Estimation in Networks / 39
1.6.4 Equitable Bandwidth Allocation for Content Distribution / 39
1.6.5 Equitable Location of Emergency Facilities / 39
1.6.6 Other Applications / 39

2 Nonlinear Resource Allocation 41
2.1 Formulation and Optimality Properties / 42
2.2 Algorithms / 48
   2.2.1 The Activity Deletion Algorithm / 48
   2.2.2 The Activity Addition Algorithm / 53
   2.2.3 The Constraints Evaluation Algorithm / 55
   2.2.4 Lower and Upper Bounds / 58
2.3 Nonlinear Resource-Usage Constraint / 58
   2.3.1 Formulation and Optimality Properties / 59
   2.3.2 Algorithms / 62
2.4 Multiple Resource Constraints: A Special Case / 66
2.5 Concluding Remarks and Literature Review / 73
Exercises / 75

3 Equitable Resource Allocation: Lexicographic Minimax and Maximin Optimization 77
3.1 Formulation and Optimality Properties / 78
3.2 Minimax Algorithms / 84
   3.2.1 The Minimax Activity Deletion Algorithm / 84
   3.2.2 The Minimax Activity Addition Algorithm / 90
   3.2.3 The Minimax Constraints Evaluation Algorithm / 94
   3.2.4 Lower and Upper Bounds / 97
3.3 The Lexicographic Minimax Algorithm / 98
3.4 Extension to Nonseparable Objective Function / 107
3.5 Concluding Remarks and Literature Review / 116
Exercises / 120

4 Equitable Resource Allocation with Substitutable Resources 123
4.1 Representations of Substitutable Resources / 124
   4.1.1 Transitive Substitutable Resources Represented by Trees / 124
CONTENTS

4.1.2 Transitive Substitutable Resources Represented by Acyclic Graphs / 125
4.1.3 Nontransitive Substitutable Resources Represented by Bipartite Graphs / 127
4.1.4 Activity-Dependent Substitutable Resources Represented by Bipartite Graphs / 128
4.1.5 Solution Approach / 129

4.2 Transitive Substitutable Resources Represented by Trees / 131
4.2.1 Formulation / 131
4.2.2 The Minimax Algorithm / 134
4.2.3 The Lexicographic Minimax Algorithm / 143
4.2.4 Lower and Upper Bounds / 151

4.3 Transitive Substitutable Resources Represented by Acyclic Graphs / 153
4.3.1 Formulation / 154
4.3.2 The Feasibility Problem / 155
4.3.3 The Minimax Algorithm / 161
4.3.4 The Lexicographic Minimax Algorithm / 165

4.4 Activity-Dependent Substitutable Resources Represented by Bipartite Graphs / 172
4.4.1 Formulation / 173
4.4.2 The Minimax Algorithm / 175
4.4.3 The Lexicographic Minimax Algorithm / 179

4.5 Concluding Remarks and Literature Review / 180
Exercises / 181

5 Multiperiod Equitable Resource Allocation 183

5.1 Formulation for Storable Resource Allocation / 184
5.2 Minimax Algorithms for Storable Resources / 187
  5.2.1 The Search-Based Algorithm / 188
  5.2.2 The Transformation-Based Algorithm / 192
  5.2.3 The Multiperiod Activity Deletion Algorithm: A Special Case / 200
5.3 The Lexicographic Minimax Algorithm / 203
5.4 Allocation of Nonstorable Resources / 210
5.5 Multiperiod Allocation of Substitutable Resources / 213
5.6 Concluding Remarks and Literature Review / 218
Exercises / 219
Learn from yesterday, live for today, hope for tomorrow.
—Albert Einstein

PERSPECTIVE

Resource allocation problems focus on the allocation of limited resources among competing activities with the intent of optimizing an objective function. This book covers a large variety of resource allocation models with special mathematical structures, solvable by elegant, efficient algorithms that take advantage of these structures. In particular, the book considers models that attempt to allocate limited resources equitably (fairly) among all competing activities. The first well-known paper in the twentieth century on resource allocation models with a special mathematical structure was published by Koopman in 1953 under the title *The Optimum Distribution of Effort*. This paper has inspired many researchers to examine related resource allocation problems.

My interest in equitable resource allocation started in the mid-1980s when a severe worldwide shortage of integrated circuits had a significant adverse effect on manufacturers of high-tech products. As was often the case at AT&T Labs/Bell Laboratories, I had a hallway chat with my colleague, Bob Smith, discussing how the effect of such shortages can be minimized. This chat evolved to more serious discussions and led in 1986 to our first paper on the topic: “Resource Allocation Among Competing Activities: A Lexicographic Minimax Approach.” Although the title is somewhat intimidating, the approach can be explained in an intuitively appealing way. Suppose each activity is associated with a performance function that depends on the level assigned to the activity. The function value may represent cost, shortage, delay, and so on, and thus, the smaller the value is, the better-off the activity. The level assigned to the activities is constrained by the available resources. In the manufacturing of high-tech products example, activities may represent circuit boards, resources are components like integrated circuits, and the performance functions represent shortages. A lexicographic minimax objective extends the well-known minimax objective and computes:
the smallest feasible performance function value for activities with the largest (i.e., worst) performance function value (this is the minimax solution), followed by

the smallest feasible performance function value for activities with the second largest (i.e., second worst) performance function value without increasing the largest value, followed by

the smallest feasible performance function value for activities with the third largest (i.e., third worst) performance function value without increasing the two largest values, and so forth.

Likewise, a lexicographic maximin objective extends the well-known maximin objective, where each performance function value may represent revenues, throughput, service quality, and so on. A lexicographic maximin objective computes

the largest feasible performance function value for activities with the smallest (i.e., worst) performance function value (this is the maximin solution), followed by

the largest feasible performance function value for activities with the second smallest (i.e., second worst) performance function value, without decreasing the smallest value, and so forth.

As a further incentive, a lexicographic minimax (or maximin) solution has been shown to satisfy desired properties of equitability (fairness), while also being pareto-optimal (efficient). Indeed, it is sometimes referred to as the “most equitably efficient solution.” Recall that a pareto-optimal solution simply means that there is no other solution where some performance function can be improved without degrading the value of some other performance function. Computing a lexicographic minimax (or maximin) solution is quite intriguing since we do not know a priori which activity would be the worst-off, the second worst-off, and so forth.

In the description above, we have assumed that a performance function is associated with a single activity. More difficult problems arise when each performance function depends on levels of multiple activities. As an example, in communication networks, a performance function may represent the throughput between a pair of nodes, while the throughput is the sum of flows across multiple paths between the node pair. Indeed, computing lexicographic minimax solutions to such problems is even more challenging. Moving along, in various applications, such as in emergency facility location and network design problems, the formulations include integer decision variables. Computing lexicographic minimax solutions to such problems requires new, innovative approaches.

Equitable resource allocation models have been found valuable in many application areas, such as logistics (e.g., product distribution and manu-
facturing), communication and computer networks, emergency facility loca-
tions, air traffic management, water rights allocation, health management,
environmental topics, and military applications. Significant work on equitable
resource allocation, scattered over many journals, has been published by many
researchers and practitioners in recent decades. My research activities have
benefited tremendously from joint efforts with my colleagues and from the
work of others.

OUTLINE OF THE BOOK

Chapter 1 introduces the reader to the topic of equitable resource allocation.
It provides a basic understanding of the methodology and describes a sample
of models in different application areas. The chapter examines various notions
of fairness and presents a detailed outline of the book.

Chapter 2 covers well-studied problems of maximizing the sum of concave
functions, where each of these functions depends on a single decision variable,
subject to a single resource constraint. The resource constraint may be linear
or nonlinear, and the decision variables are continuous. Although such models
do not provide equitable solutions, the algorithms presented are quite instruc-
tive and serve as a worthwhile lead into equitable resource allocation models.

Chapters 3–6 focus on equitable resource allocation models with continu-
ous decision variables, and Chapter 7 presents models with integer decision
variables.

Chapter 3 first covers a basic equitable resource allocation problem with
a lexicographic minimax separable objective function, where each perfor-
mance function depends on a single activity level, subject to multiple resource
constraints. For certain classes of performance functions, the optimal solutions
are obtained by manipulating closed-form expressions; thus, large-scale prob-
lems can be solved with a small computational effort. The model is then
extended to nonseparable objective functions, where a performance function
may depend on multiple activity levels. The algorithms are more involved; for
the special case of linear performance functions, the lexicographic minimax
(maximin) solution is obtained by solving a sequence of linear programming
problems.

Chapter 4 extends the models by considering substitutable resources.
Consideration of substitutions among resources adds significant flexibility to
resource allocation. It is particularly important in a dynamic environment
with rapidly changing technologies. For example, in high-tech manufacturing,
subsets of components, such as integrated circuits, are often substitutable. The
chapter presents various models with different degrees of flexibility among
substitutable resources and provides efficient algorithms that solve these
models.

Chapter 5 extends the models to a multiperiod setting. Much of the material
examines storable resources, where resources not used in one period can be
used in subsequent periods. Examples for such resources include nonperishable commodities such as high-tech components, water reservoirs, and oil reserves. The chapter also examines models with nonstorable resources and models that consider substitutable resources in a multiperiod setting.

Chapter 6 examines equitable allocation of network resources. Network flow models with a single fixed path per demand between each node pair and network flow models with multiple paths per demand are examined. The similarities and differences between these models and those presented in Chapter 3 are emphasized. The chapter also examines content distribution of multiple programs, applicable, for example, to video-on-demand applications. The algorithms determine optimal bandwidth allocations along the links of a network comprised of multiple tree topologies, where programs are broadcast from the roots of these trees.

Chapter 7 covers equitable resource allocation problems with integer decision variables. Unlike in all problems discussed in previous chapters, the feasible region is not convex, leading to significantly new challenges and more complicated algorithms. In general, a sequence of mixed integer programming problems needs to be solved with added auxiliary variables and constraints. Examples of applications from different areas, including allocation of seats in a legislative body, emergency facility locations, sensor locations, and communication networks, are described, and various solution methods are presented.

PURPOSE OF THE BOOK

Significant work, scattered over many journals, has been published in recent decades on equitable resource allocation. Hence, it seems quite timely to unify the results in a single book on this topic. The presentation in this book attempts to provide a unified approach that evolves from simple models to more complex ones. The book presents a comprehensive exposition of models, algorithms, and applications intended to enhance understanding of and provide insight into the topic. It is expected to be valuable to people with undergraduate and graduate degrees in engineering, operations research, computer science, and applied math. It assumes some basic familiarity with optimization methodologies such as linear programming. Nevertheless, the material is presented using both intuitive and rigorous explanations.

The book is targeted at two communities. The first includes instructors and students and the second includes scientists and practitioners.

Instructors and Students

The book is designed as a one-semester textbook for graduate and postgraduate students and advanced undergraduates. As mentioned above, some familiarity with optimization methodologies is desired. For graduate and advanced undergraduate students, the emphasis can be on models, intuitive explanations
of algorithms, and applications. For postgraduate students, this material can be enhanced with more rigorous analysis and a deeper understanding of algorithms. The numerical examples throughout the book, as well as the exercises at the end of Chapters 2–7, are expected to help explain the material. Time permitting, the book can be supplemented by some of the references. Note that each chapter has a final section of concluding remarks and a discussion of references.

The material in this book can also be used as part of a course on communication network design or on modeling and optimization methodologies. The core of such a module may consist of the introduction in Chapter 1 and selected material from Chapter 3. Selection of additional material will depend on the course orientation. For instance, in a course on communication networks, selected material from Chapters 6 and 7 might be of interest.

**Scientists and Practitioners**

Professionals interested in optimal resource allocation problems in diverse application areas will find this book to be a valuable reference on their shelves. Scientists will find ample ideas for exploring new models and methodologies. Practitioners will find clear presentations of models and algorithms that can be readily adapted and modified to problems they may encounter. Often, the algorithms described will be integrated as an important module that is repeatedly solved within a more complex system. Presentation of the algorithms includes step-by-step instructions, which should make the translation into executable software relatively easy.

**Hanan Luss**
I was fortunate to work for 25 years at AT&T Labs/Bell Laboratories and 12 years at Telcordia Technologies (formerly Bellcore) in environments that encouraged synergy between research and applied work. These environments have had a major influence on my thinking about problems, which will be evident throughout this book. I have interacted with numerous very talented people and have learned a great deal from them. We have collaborated on research problems and applied work, covering different topics including resource allocation, communication network design, logistics, and capacity planning. I would like to thank all my colleagues for the productive and fun time we spent exploring and solving various problems. At the risk of omitting many deserving names, I would like to mention a few of my colleagues who influenced my thinking, in particular (in alphabetic order), Tami Carpenter, Linos Frantzeskakis, Mohan Gawande, Heinz Groeflin, Oktay Gunluk, Shiv Gupta (my PhD advisor), Shlomo Halfin, Rachelle Klein, John Klincewicz, Murali Kodialam, K.R. Krishnan, Chuck McCallum, Marc Meketon, Beth Munson, Eric Rosenberg, Moshe Rosenwein, Uriel Rothblum, Bob Smith, Andrew Vakhutinsky, Ward Whitt, and Richard Wong.

Special gratitude is due to my colleague and friend Russell Hsing, coeditor of the Information and Communication Technology Series. His continuous encouragement during the preparation of this book has been invaluable. It has also been a pleasure to work with Diana Gialo and Simone Taylor of John Wiley & Sons and with Stephanie Sakson of Toppan Best-set Premedia. Their excellent support during this project is very much appreciated.

I thank my dear wife, Dalia, the light of my life, for the great journey we have had for five decades, and more to come. This book is dedicated to her. Last, but not least, I am very grateful to my son, Ronny Luss. Despite his busy schedule as a postdoc, he carefully read the entire manuscript and provided numerous comments and valuable suggestions. His efforts significantly enhanced the presentation and reduced the number of inaccuracies and errors. Obviously, any remaining inaccuracies and errors are fully my responsibility.

H.L.
Golf is a game whose aim is to hit a very small ball into an even smaller hole, with weapons singularly ill-designed for that purpose.

—Winston Churchill

Over 70 years have passed since the emergence of operations research during World War II. During this relatively short period, operations research has contributed significantly to diverse areas in the military, industry, and government, including logistics, communication, transportation, energy, health care, manufacturing, marketing, finance, and more. A significant part of operations research focuses on allocating limited resources among competing activities, or to put it simply, how to allocate the cake among the cake lovers (Fig. 1.1).

This chapter introduces the reader to certain classes of resource allocation models for which elegant and efficient solution methodologies have been developed, and which have been found to be valuable in diverse application areas.

1.1 PERSPECTIVE

Resource allocation problems focus on the allocation of limited resources among competing activities with the intent of optimizing an objective function.
Initially, during World War II, solution methodologies were developed to support critically important military activities including deployment of radar systems, antisubmarine warfare, and bombing strategies. After the war, these methodologies were welcomed enthusiastically in order to help solve problems across diverse application areas in the public and private sectors. Major companies in the telecommunication, oil, transportation, automobile, high-tech, and other sectors established operations research groups to solve major recurring resource allocation problems in support of strategic and tactical problems. Government agencies used these methodologies to address important societal issues such as those occurring in health care, education, water resources, and environmental topics.

It is no doubt that linear programming has been, and still is, the most celebrated methodology used to solve resource allocation problems. Kantorovich, Dantzig, and von Neumann are regarded as the founders of linear programming, and the simplex method for solving such problems was published by Dantzig in 1947. Linear programming models consist of either minimizing or maximizing a linear objective function while satisfying linear constraints. Major advances in linear programming methodologies and
increased computing power have facilitated solving very large problems with hundreds of thousands and even millions of decision variables and constraints.

This book covers a large variety of resource allocation models with special mathematical structures, solvable by elegant, efficient algorithms that take advantage of these structures. Moreover, the book primarily considers models that attempt to allocate the limited resources equitably (fairly) among competing activities; the notion of such equitable allocation will be described in the next section. From a historical perspective, the first well-known paper on resource allocation models with a special mathematical structure was published by Koopman in 1953 under the title of *Optimum Distribution of Effort*. Koopman followed up with a series of three papers on the theory of search, where the third of these papers presents a solution methodology for optimal distribution of searching effort. These seminal papers mark the beginning of the topic of resource allocation models with special mathematical structures.

### 1.2 EQUITABLE RESOURCE ALLOCATION: LEXICOGRAPHIC MINIMAX (MAXIMIN) OPTIMIZATION

Consider a resource allocation problem where multiple resources are allocated among numerous activities. We use the following notation:

**Indices and Sets**

- $i$ = Index for resources.
- $j$ = Index for activities.
- $I$ = Set of resources; $I = \{1, 2, \ldots, m\}$.
- $J$ = Set of activities; $J = \{1, 2, \ldots, n\}$.

**Parameters**

- $b_i$ = Amount available of resource $i$; $b_i > 0$ for all $i \in I$.
- $a_{ij}$ = Amount of resource $i$ consumed by a single unit of activity $j$; $a_{ij} \geq 0$ for all $i \in I$ and $j \in J$, at least one $a_{ij} > 0$ for each $i \in I$, and at least one $a_{ij} > 0$ for each $j \in J$.
- $l_j$ = Lower bound ($l_j \geq 0$) on the selected level for activity $j$ for all $j \in J$.
- $u_j$ = Upper bound ($u_j \geq l_j$) on the selected level for activity $j$ for all $j \in J$.

**Decision Variables**

- $x_j$ = Activity level selected for activity $j$ for all $j \in J$; $x = \{x_j; j \in J\}$.

**Performance Functions**

- $f_j(x_j) = \text{Performance function for activity } j \text{ for all } j \in J$. 
The resource allocation problem attempts to find activity levels that optimize some objective function while satisfying the resource constraints. These constraints are formulated as follows:

\[ \sum_{j \in J} a_{ij} x_j \leq b_i \text{ for all } i \in I, \quad (1.2.1a) \]
\[ l_i \leq x_j \leq u_j \text{ for all } j \in J. \quad (1.2.1b) \]

Note that resource constraints (1.2.1a) are restricted to having all parameters \( a_{ij} \geq 0 \) and all inequalities as “\( \leq \).” Such resource constraints are also referred to as knapsack constraints.

Now, suppose that each activity \( j \) produces a value of \( r_j > 0 \) per activity unit, which implies a linear performance function \( f_j(x_j) = r_j x_j \). Performance functions that are strictly increasing with their assigned activity level may represent revenues, profits, service characteristics, throughput, and so on. A linear programming model then attempts to maximize \( \sum_{j \in J} r_j x_j \). Suppose \( I \) includes 100 resources and \( J \) includes 1000 activities, all lower bounds are zero, and all upper bounds are very large. Since the optimal solution is typically at an extreme point, at most 100 activities will be assigned values above zero. In other words, in order to optimize the total value over all activities, at least 900 activity levels are fixed at zero while resources are allocated to just 100 activities. Such a disproportionate allocation scheme may not be acceptable in many applications as it may be perceived as grossly unfair (of course, such extreme examples can be avoided through the imposition of lower and upper bounds as well as other linear constraints).

This drawback can be remedied by using nonlinear performance functions for the activities. Thus, instead of maximizing \( \sum_{j \in J} r_j x_j \), we might maximize \( \sum_{j \in J} f_j(x_j) \) where the functions \( f_j(x_j) \) are strictly increasing and strictly concave. Such performance function implies that the marginal increase in \( f_j(x_j) \) decreases as \( x_j \) increases. An equivalent problem is formulated with performance functions \( f_j(x_j) \) that are strictly decreasing convex functions, representing cost, delay, poor service, and so on. The objective function is then changed to minimizing \( \sum_{j \in J} f_j(x_j) \).

In many applications, it is important to allocate resources fairly among the activities. These include, for example, allocation of bandwidth in telecommunication networks, allocation of takeoff and landing “slots” at airports, and allocation of water resources. This gives rise to minimax (or maximin) objective functions. In models with a minimax objective function, expressed as \( \min_x [\max_{j \in J} f_j(x_j)] \), we find feasible activity levels that satisfy all constraints so that the largest (i.e., worst) performance function value is as small as possible. Consider resource allocation problems with constraints (1.2.1a) and (1.2.1b), where all resource constraints are of the knapsack type (all \( a_{ij} \geq 0 \) and all inequalities are “\( \leq \)”). In the absence of other constraints, it is reasonable to assume that, for a minimax objective, the performance functions are strictly
decreasing, or at least nonincreasing. Equivalently, in a maximin objective function, expressed as \( \max_j [\min_i f_i(x_i)] \), we find feasible activity levels so that the smallest performance function value is as large as possible. It is now reasonable to assume that the performance functions are strictly increasing or at least nondecreasing.

A resource allocation model with a minimax objective function can readily be transformed to a model with a maximin objective function and vice versa by the following identities:

\[
\min_x [\max_{j \in J} f_j(x_j)] = -\max_x [\min_{j \in J} (-f_j(x_j))] \tag{1.2.2a}
\]

and

\[
\max_x [\min_{j \in J} f_j(x_j)] = -\min_x [\max_{j \in J} (-f_j(x_j))] \tag{1.2.2b}
\]

Note that the minimax (or maximin) objective function seeks a solution with the best feasible performance function value for the worst-off activity. Unfortunately, there may be numerous feasible activity level assignments that result in the minimax (or maximin) solution. Thus, this objective function does not provide any guidance as to which solution should be selected from among all such solutions. Although a minimax solution provides some “safety net” to the activities, it may still be perceived as unfair by a majority of the activities. In addition, it is often criticized because a minimax solution is not a pareto-optimal solution. A pareto-optimal solution (also referred to as an efficient solution) is defined as a solution where no performance function value can be improved without degrading the value of some other performance function.

A natural extension of the minimax objective is within the scope of multi-objective optimization, where each performance function \( f_j(x_j) \) serves as an objective to be optimized. In this extension, we compute

- the smallest feasible performance function value for activities with the largest (i.e., worst) performance function value (this is the minimax solution), followed by
- the smallest feasible performance function value for activities with the second largest (i.e., second worst) performance function value without increasing the largest value, followed by
- the smallest feasible performance function value for activities with the third largest (i.e., third worst) performance function value without increasing the two largest values, and so forth.

Likewise, we extend the maximin objective and compute

- the largest feasible performance function value for activities with the smallest (i.e., worst) performance function value (this is the maximin solution), followed by
• the largest feasible performance function value for activities with the second smallest (i.e., second worst) performance function value, without decreasing the smallest value, and so forth.

The extended minimax and maximin objectives are called the lexicographic minimax and lexicographic maximin objectives, respectively. The resulting solution is pareto-optimal, that is, efficient. Intuitively, the solution is also perceived as equitable by all activities, and hence it is often referred to as an equitably efficient solution. Properties of equitable solutions will be described in Section 1.4.

In the present discussion, we assume that the objective function is separable; that is, performance function \( f_j \) depends only on the level \( x_j \) assigned to activity \( j \). Later in this section, we extend the discussion to a nonseparable objective function, where each performance function may depend on values assigned to multiple activities.

We now formalize the concept of a lexicographic minimax solution. We first need to define the term lexicographic. Consider two vectors \( v^1 \) and \( v^2 \), each with \( n \) elements, \( v^1 = [v_1^1, v_2^1, \ldots, v_n^1] \) and \( v^2 = [v_1^2, v_2^2, \ldots, v_n^2] \), and suppose \( v_j^1 = v_j^2 \) for \( j = 1, 2, \ldots, k \) \((k < n)\) and \( v_j^1 > v_j^2 \) for \( j = k + 1 \). Then, vector \( v^1 \) is lexicographically larger than vector \( v^2 \) (and, equivalently, vector \( v^2 \) is lexicographically smaller than vector \( v^1 \)). Consider now a feasible solution vector \( x \) to a resource allocation problem, for example, a solution that satisfies constraints (1.2.1a) and (1.2.1b) while having performance function values \( f_j(x_j) \) for all \( j \in J \). Let \( f^{(n)}(x) = [f_j(x_j), f_{j_2}(x_{j_2}), \ldots, f_{j_n}(x_{j_n})] \) be the vector of performance functions under allocation \( x \), where the elements of this vector are sorted in nonincreasing order. Thus, the vector \( f^{(n)}(x) \) is expressed as follows:

\[
\begin{align*}
\mathbf{f}^{(n)}(x) &= [f_h(x_h), f_{j_2}(x_{j_2}), \ldots, f_{j_n}(x_{j_n})], \\
\text{where} \\
f_h(x_h) &\geq f_{j_2}(x_{j_2}) \geq \cdots \geq f_{j_n}(x_{j_n}).
\end{align*}
\]

A lexicographic minimax objective function searches for a feasible vector \( x \) that provides the lexicographic smallest vector of performance functions whose elements (the performance function values) are sorted in a nonincreasing order. In other words, it searches for the lexicographically smallest feasible vector \( f^{(n)}(x) \).

We are now ready to formulate a basic resource allocation problem with a lexicographic minimax objective function, referred to as Problem L-RESOURCE ("L" stands for lexicographic minimax as well as, depending on the formulation, for lexicographic maximin). In this problem, performance function \( f_j(x_j) \) is assumed to be strictly decreasing and depends only on \( x_j \), and the constraints include only knapsack-type resource constraints and lower and
upper bound constraints. The resulting formulation is a lexicographic minimax optimization problem. The lexicographic maximin optimization problem will be discussed later.

**PROBLEM L-RESOURCE (lex-minimax objective)**

\[ V^L = \text{lexmin} \{ f^{(n)}(x) = [f_{j_1}(x_{j_1}), f_{j_2}(x_{j_2}), \ldots, f_{j_n}(x_{j_n})] \} \]  

subject to

\[ f_{j_1}(x_{j_1}) \geq f_{j_2}(x_{j_2}) \geq \cdots \geq f_{j_n}(x_{j_n}), \]  

\[ \sum_{j \in J} a_{ij} x_j \leq b_i \quad \text{for all } i \in I, \]  

\[ l_j \leq x_j \leq u_j \quad \text{for all } j \in J. \]

We assume that \( \sum_{j \in J} a_{ij} l_j \leq b_i \) for all \( i \in I \), which implies that a feasible solution exists. Furthermore, since all \( a_{ij} \geq 0 \) and at least one \( a_{ij} > 0 \) for each \( j \in J \), resource constraints (1.2.4c) imply that the solution is bounded even without the upper bounds in (1.2.4d). We use throughout the book superscript \( L \) to denote optimal values for problems with a lexicographic minimax (or lexicographic maximin) objective function. We often refer to these values as lexicographic minimax (or lexicographic maximin) values. Likewise, we use superscript * to denote optimal values for problems with a minimax (or maximin) objective function, and often refer to these values as minimax (or maximin) values. Objective function (1.2.4a) lexicographically minimizes the vector \( f^{(n)}(x) \), where constraints (1.2.4b) enforce the appropriate order of the elements of this vector. Note that the lexicographic minimax objective is quite different than a standard lexicographic optimization objective where the order in which performance functions are optimized is given as input. Here, the order is unknown as it must satisfy constraints (1.2.4b). Constraints (1.2.4c) are knapsack resource constraints, and constraints (1.2.4d) enforce lower and upper bound values for all activity levels. We will also write, on occasion,

\[ V^L = \text{lex-minimax} \{ f(x) = [f_1(x_1), f_2(x_2), \ldots, f_n(x_n)] \} \]  

instead of (1.2.4a) and (1.2.4b). Here, \( f(x) \) is the unsorted vector of performance functions. Expressing the lexicographic minimax objective by (1.2.5) is more convenient in numerical examples.

Note that Problem L-RESOURCE, as formulated by (1.2.4a)–(1.2.4d), or by (1.2.5), (1.2.4c), and (1.2.4d), is not a standard formulation for a mathematical optimization problem. However, as will be demonstrated throughout the
book, particularly in Chapters 3–6, lexicographic minimax solutions for many problems, including Problem L-RESOURCE, can be obtained by repeatedly solving problems with a minimax objective function subject to the same constraints with minor modifications. These minimax problems can readily be formulated as standard optimization problems. On the other hand, as will be shown in Chapter 7, computing lexicographic minimax solutions for problems with integer decision variables is, in general, much more difficult as it requires adding many auxiliary variables and constraints.

As stated earlier, a lexicographic minimax solution to Problem L-RESOURCE can be characterized quite intuitively as follows:

(a) It provides the smallest feasible performance function value for activities with the largest performance function value, followed by the smallest feasible performance function value for activities with the second largest performance function value without increasing the largest value, followed by the smallest feasible performance function value for activities with the third largest performance function value without increasing the two largest values, and so forth.

A precise mathematical characterization for Problem L-RESOURCE will be presented in Chapter 3. Property (a) is the essence of lexicographic minimax optimization, not just for Problem L-RESOURCE, and does not require any assumptions regarding the performance functions or the feasible region. Characterization (a) is simply an alternate definition of providing the smallest lexicographic vector whose elements, the performance function values, are sorted in a nonincreasing order.

Now, suppose that the performance functions $f_j(x_j)$ are strictly decreasing and continuous. Then, a lexicographic minimax solution to Problem L-RESOURCE also satisfies the following properties:

(b) No performance function value can be feasibly decreased without increasing the performance function value of some other activity whose performance function value is already at least as large.

(c) No activity level can be feasibly increased without decreasing the level of some other activity whose performance function value is already at least as large.

Many of the problems presented in Chapters 3–6 have a lexicographic minimax separable objective function, where the performance functions $f_j(x_j)$ are strictly decreasing and continuous. These problems have mathematical structures that allow for finding the lexicographic minimax solution by repeatedly solving minimax problems of the same format. Such algorithms proceed as follows: A minimax problem is solved after which some activity levels are fixed at their lexicographic minimax value. A new minimax problem is then formulated without these activities and with only leftover resources. A key concept for
making this approach work is the minimal solution for a minimax problem, defined as follows: Suppose optimal solution \( x^* \) is the minimal solution. Then \( x^* \leq y^* \) component-wise, where \( y^* \neq x^* \) is any other optimal solution to the minimax problem. Clearly, \( x^*_j < y^*_j \) for some \( j \in J \). After a minimax problem is solved, some activity levels are fixed at their minimal values. Since the resource constraints are of the knapsack type, the amount of leftover resources for the subsequent minimax problem is the largest possible. This procedure is repeated until all activity levels are fixed. The fixed values of all activities comprise the lexicographic minimax solution. In the description above, resources are assigned to fixed activities after a minimax problem is solved. However, in some problems (e.g., as in Section 4.3), assignment of resources to activities must be deferred until all activity levels are fixed at their lexicographic minimax value. Furthermore, when the objective function is not separable, the lexicographic minimax solution is still obtained by repeatedly solving minimax problems, but, as will be seen in Section 3.4, the algorithm is more complicated.

We illustrate the solution approach by considering Problem L-RESOURCE with a lexicographic minimax objective and strictly decreasing performance functions. We assume bounds \( l_j = 0 \) and \( u_j = \infty \) for all \( j \in J \). Consider finding the minimal solution to the minimax problem

\[
V^* = \min \left[ \max_{j \in J} f_j(x_j) \right] 
\]

subject to

\[
\sum_{j \in J} a_{ij} x_j \leq b_i \text{ for all } i \in I, 
\]

\[
x_j \geq 0 \text{ for all } j \in J. 
\]

This minimax problem can readily be formulated as a standard optimization problem by replacing objective function (1.2.6a) with the objective \( V^* = \min_x V \) and adding constraints \( V \geq f_j(x_j) \) for all \( j \in J \). Figure 1.2 presents resource constraints (1.2.6b) for the \( m \) resources (rows) and the \( n \) activities (columns). Since the performance functions are strictly decreasing, there is at least one resource constraint that is satisfied at equality by any optimal solution. Suppose that the minimal solution has resource \( i_c \) as the single critical resource that is fully used, that is, this constraint is satisfied at equality. The symbol + in row \( i_c \) means that the corresponding \( a_{i_c j} > 0 \) while a zero indicates that the corresponding \( a_{ij} = 0 \) (\( a_{ij} \geq 0 \) for all \( i \in I \) and \( j \in J \)). As will be proven in Chapter 3, the lexicographic minimax values of variables associated with \( a_{i_c j} > 0 \) are equal to their value at the minimal solution to the minimax problem, where the minimal solution satisfies \( f_j(x^*_j) = V^* \) if \( x^*_j > 0 \) and \( f_j(x^*_j) \leq V^* \) if \( x^*_j = 0 \). Hence, all activities in the shaded blocks in Figure 2.1 are fixed and deleted from the formulation of the next minimax problem. Also, resource \( i_c \) is deleted, while all other \( b_i 's \) are updated to account for resources used by the deleted
activities. A new minimal solution is then found to the reduced minimax problem. This repeated determination of minimal solutions to minimax problems is continued until all variables are fixed at their optimal values. Formal development of the methodology to solve this problem is deferred to Chapter 3. It should, however, be clear that the key to developing a computationally efficient algorithm for solving lexicographic minimax resource allocation problems, like Problem L-RESOURCE, is the development of efficient algorithms to the underlying minimax problems. As will be seen, for certain classes of performance functions, each of the minimax problems can be solved by manipulating closed-form expressions, resulting in extremely efficient algorithms. For other performance functions, more intensive computations will be required.

This procedure is further illustrated by the following example:

\[
V^L = \text{lex-minimax}\{f(x) = [20 - 2x_1, 15 - x_2, 10 - x_3]\}
\]

subject to

\[
\begin{align*}
    x_1 + x_2 & \leq 4, \\
    x_1 + x_3 & \leq 5, \\
    x_j & \geq 0, \ j = 1, 2, 3.
\end{align*}
\]