1

INTRODUCTION

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 to 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.

Equitable Resource Allocation: Models, Algorithms, and Applications, First Edition. Hanan Luss.

Figure 1.1 Resource allocation: allocating the cake among people with different needs.

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 $j \in J$; $\boldsymbol{x} = \{x_j: j \in J\}$.

Performance Functions

$f_j(x_j)$ = Performance function for activity j 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:

$$\sum_{j \in J} a_{ij} x_j \leq b_i \text{ for all } i \in I, \tag{1.2.1a}$$

$$l_j \leq x_j \leq u_j \text{ for all } j \in J. \tag{1.2.1b}$$

Note that resource constraints (1.2.1a) are restricted to having all parameters $a_{ij} \geq 0$ and all inequalities as "≤." 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 "≤"). 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 $\max_{\boldsymbol{x}}[\min_{j\in J} f_j(x_j)]$, 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:

$$\min_{\boldsymbol{x}}[\max_{j\in J} f_j(x_j)] = -\max_{\boldsymbol{x}}[\min_{j\in J}(-f_j(x_j))] \tag{1.2.2a}$$

and

$$\max_{\boldsymbol{x}}[\min_{j\in J} f_j(x_j)] = -\min_{\boldsymbol{x}}[\max_{j\in J}(-f_j(x_j))]. \tag{1.2.2b}$$

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 multiobjective 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 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 $\boldsymbol{v}^1$ and $\boldsymbol{v}^2$, each with n elements, $\boldsymbol{v}^1 = [v_1^1, v_2^1, \ldots, v_n^1]$ and $\boldsymbol{v}^2 = [v_1^2, v_2^2, \ldots, v_n^2]$, 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 $\boldsymbol{v}^1$ is lexicographically larger than vector $\boldsymbol{v}^2$ (and, equivalently, vector $\boldsymbol{v}^2$ is lexicographically smaller than vector $\boldsymbol{v}^1$). Consider now a feasible solution vector $\boldsymbol{x}$ 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 $\boldsymbol{f}^{(n)}(\boldsymbol{x}) = [f_{j_1}(x_{j_1}), f_{j_2}(x_{j_2}), \ldots, f_{j_n}(x_{j_n})]$ be the vector of performance functions under allocation $\boldsymbol{x}$, where the elements of this vector are sorted in nonincreasing order. Thus, the vector $\boldsymbol{f}^{(n)}(\boldsymbol{x})$ is expressed as follows:

$$\boldsymbol{f}^{(n)}(\boldsymbol{x}) = [f_{j_1}(x_{j_1}), f_{j_2}(x_{j_2}), \ldots, f_{j_n}(x_{j_n})], \tag{1.2.3a}$$

where

$$f_{j_1}(x_{j_1}) \geq f_{j_2}(x_{j_2}) \geq \cdots \geq f_{j_n}(x_{j_n}). \tag{1.2.3b}$$

A lexicographic minimax objective function searches for a feasible vector $\boldsymbol{x}$ 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 $\boldsymbol{f}^{(n)}(\boldsymbol{x})$.

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)

$$\boldsymbol{V}^L = \operatorname*{lexmin}_{\boldsymbol{x}}\{\boldsymbol{f}^{(n)}(\boldsymbol{x}) = [f_{j_1}(x_{j_1}), f_{j_2}(x_{j_2}), \ldots, f_{j_n}(x_{j_n})]\} \tag{1.2.4a}$$

subject to

$$f_{j_1}(x_{j_1}) \geq f_{j_2}(x_{j_2}) \geq \cdots \geq f_{j_n}(x_{j_n}), \tag{1.2.4b}$$

$$\sum_{j \in J} a_{ij} x_j \leq b_i \text{ for all } i \in I, \tag{1.2.4c}$$

$$l_j \leq x_j \leq u_j \text{ for all } j \in J. \tag{1.2.4d}$$

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 $\boldsymbol{f}^{(n)}(\boldsymbol{x})$, 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,

$$\boldsymbol{V}^L = \operatorname*{lex\text{-}minimax}_{\boldsymbol{x}}\{\boldsymbol{f}(\boldsymbol{x}) = [f_1(x_1), f_2(x_2), \ldots, f_n(x_n)]\} \tag{1.2.5}$$

instead of (1.2.4a) and (1.2.4b). Here, $\boldsymbol{f}(\boldsymbol{x})$ 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 $\boldsymbol{x}^*$ is the minimal solution. Then $\boldsymbol{x}^* \leq \boldsymbol{y}^*$ component-wise, where $\boldsymbol{y}^* \neq \boldsymbol{x}^*$ 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

$$V^* = \min_{\boldsymbol{x}}[\max_{j \in J} f_j(x_j)] \tag{1.2.6a}$$

subject to

$$\sum_{j \in J} a_{ij} x_j \leq b_i \text{ for all } i \in I, \tag{1.2.6b}$$

$$x_j \geq 0 \text{ for all } j \in J. \tag{1.2.6c}$$

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 $a_{i_c j} = 0$ ($a_{ij} \geq 0$ 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

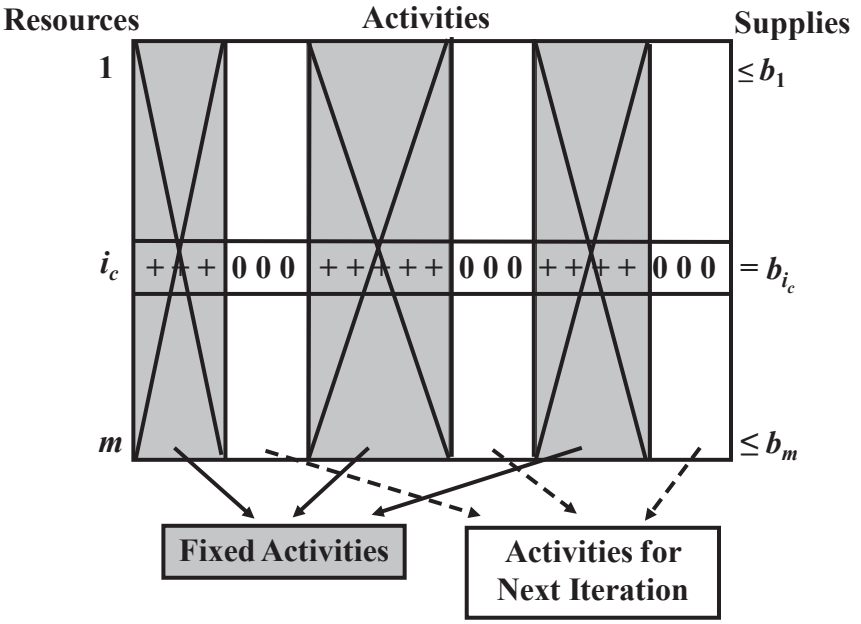

Figure 1.2 A critical resource constraint in the minimax problem.

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:

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

subject to

$$x_1 + x_2 \leq 4,$$

$$x_1 + x_3 \leq 5,$$

$$x_j \geq 0,\ j = 1, 2, 3.$$

The solution of the first minimax problem is $V^* = 14$ with the minimal solutions $x_1^* = 3, x_2^* = 1$, and $x_3^* = 0$. The first constraint is fully used so that activities 1 and 2 are fixed at their lexicographic minimax values $x_1^L = x_1^* = 3$ and $x_2^L = x_2^* = 1$ and the first constraint is deleted. The second minimax problem is formulated as $V^* = \text{minimax}_{x_3}[10 - x_3]$ subject to $x_3 \le 5 - 3$ and $x_3 \ge 0$, implying $V^* = 8$ and $x_3^* = 2$. The lexicographic minimax solution is therefore obtained as $\boldsymbol{V}^L = \boldsymbol{f}^{(3)}(\boldsymbol{x}^L) = [14, 14, 8]$ with optimal activity levels $x_1^L = 3, x_2^L = 1$, and $x_3^L = 2$.

Although much of the discussion presented so far was in terms of lexicographic minimax optimization problems, these problems can readily be translated to lexicographic maximin optimization problems. Let $w_j(x_j)$ be the performance function defined as $w_j(x_j) = -f_j(x_j)$ for all $j \in J$. Let $\boldsymbol{w}^{(n)}(\boldsymbol{x}) = [w_{j_1}(x_{j_1}), w_{j_2}(x_{j_2}), \ldots, w_{j_n}(x_{j_n})]$ be the vector performance functions under allocation $\boldsymbol{x}$, where the elements of this vector are sorted in nondecreasing order (note the reversal of the sorting). Formally, the vector $\boldsymbol{w}^{(n)}(\boldsymbol{x})$ is expressed as follows:

$$\boldsymbol{w}^{(n)}(\boldsymbol{x}) = [w_{j_1}(x_{j_1}), w_{j_2}(x_{j_2}), \ldots, w_{j_n}(x_{j_n})], \tag{1.2.7a}$$

where

$$w_{j_1}(x_{j_1}) \le w_{j_2}(x_{j_2}) \le \cdots \le w_{j_n}(x_{j_n}). \tag{1.2.7b}$$

Formulation of Problem L-RESOURCE as a *lexicographic maximin optimization problem* is as follows (here "L" stands for lexicographic maximin).

PROBLEM L-RESOURCE (lex-maximin objective)

$$\boldsymbol{W}^L = \underset{\boldsymbol{x}}{\text{lexmax}}\{\boldsymbol{w}^{(n)}(\boldsymbol{x}) = [w_{j_1}(x_{j_1}), w_{j_2}(x_{j_2}), \ldots, w_{j_n}(x_{j_n})]\} \tag{1.2.8a}$$

subject to

$$w_{j_1}(x_{j_1}) \le w_{j_2}(x_{j_2}) \le \cdots \le w_{j_n}(x_{j_n}), \tag{1.2.8b}$$

$$\sum_{j \in J} a_{ij} x_j \le b_i \text{ for all } i \in I, \tag{1.2.8c}$$

$$l_j \le x_j \le u_j \text{ for all } j \in J. \tag{1.2.8d}$$

The optimal activity levels in Problem L-RESOURCE with a lexicographic minimax objective and in Problem L-RESOURCE with a lexicographic maximin objective when $w_j(x_j) = -f_j(x_j)$ for all $j \in J$ are identical and the objective values satisfy $\boldsymbol{W}^L = -\boldsymbol{V}^L$. Throughout the book, we will use both lexicographic minimax and lexicographic maximin formulations. Transformation from one formulation to the other and the minor changes needed in the corresponding algorithms are quite obvious.

Again, we will also write, on occasion,

$$\boldsymbol{W}^L = \text{lex-}\max_{\boldsymbol{x}}\text{imin}\{\boldsymbol{w}(\boldsymbol{x}) = [w_1(x_1), w_2(x_2), \ldots, w_n(x_n)]\} \tag{1.2.9}$$

instead of (1.2.8a) and (1.2.8b). Here, $\boldsymbol{w}(\boldsymbol{x})$ is the unsorted vector of performance functions.

Note that so far, the objective function is assumed to be separable; that is, performance function j depends only on x_j. Thus, the set of activities J also represents the set of performance functions. In more general cases, the objective function is nonseparable where each performance function may depend on a subset of activities $j \in J$, referred to as the *effective activity*. For example, a performance function value may be determined by a linear combination of the activity levels in the corresponding subset, referred to as the *effective activity level*. We continue to use j as an index for activities and introduce new notation:

d = Index for effective activities and the corresponding performance functions.
D = Set of effective activities and the corresponding performance functions; $D = \{1, 2, \ldots, n\}$.
J_d = Set of activities that contribute to effective activity d for all $d \in D$.

The vector of activity levels that contribute to effective activity d is denoted as $\boldsymbol{x}_d = \{x_j: j \in J_d\}$, and the vector of all activity levels is denoted as $\boldsymbol{x} = \{x_j: j \in \cup_{d\in D} J_d\}$. The performance function d may, for example, be $f_d(\boldsymbol{x}) = \sum_{j\in J_d} \lambda_{dj} x_j$ for each $d \in D$, where the λ_{dj}'s are positive parameters. For the lexicographic minimax objective, the vector $\boldsymbol{f}^{(n)}(\boldsymbol{x})$ is then expressed as follows:

$$\boldsymbol{f}^{(n)}(\boldsymbol{x}) = [f_{d_1}(\boldsymbol{x}_{d_1}), f_{d_2}(\boldsymbol{x}_{d_2}), \ldots, f_{d_n}(\boldsymbol{x}_{d_n})], \tag{1.2.10a}$$

where

$$f_{d_1}(\boldsymbol{x}_{d_1}) \geq f_{d_2}(\boldsymbol{x}_{d_2}) \geq \cdots \geq f_{d_n}(\boldsymbol{x}_{d_n}). \tag{1.2.10b}$$

Consider Problem L-RESOURCE with a nonseparable objective function where (1.2.4a) and (1.2.4b) are replaced by (1.2.10a) and (1.2.10b), and suppose $f_d(\boldsymbol{x}) = \sum_{j\in J_d} \lambda_{dj} x_j$ for each $d \in D$. A lexicographic minimax solution to this problem provides the smallest feasible performance function value for effective activities with the largest performance function value, followed by the smallest feasible performance function value for effective activities with the second largest performance function value without increasing the largest value, and so forth. However, even for linear performance functions, solving the lexicographic minimax problems requires solving repeatedly minimax problems formulated as linear programming problems. Moreover, as will be seen, once a minimax problem is solved, the identification of a critical resource

constraint that is satisfied at equality cannot readily be used to fix activity levels. Instead, *saturated effective activities* need to be identified. A saturated effective activity is defined as one whose corresponding performance function value cannot be further decreased without adversely affecting other performance functions, whose value is at least as large. While saturated effective activities are excluded from the objective function of subsequent minimax problems, their saturation levels are protected through newly added constraints. In contrast, for certain classes of performance functions, including linear functions, Problem L-RESOURCE with a separable objective function is solved more easily by manipulating closed-form expressions.

We next present an example of an uncapacitated facility location problem for emergency services, where it is important to provide equitable services to all locations. The resource constraint here limits the number of facilities that can be placed. The example highlights the need for a lexicographic minimax objective rather than simply a minimax objective. Consider a network where we seek to open an emergency facility (e.g., a fire station) at one of the nodes so that the distance from the farthest node to the facility is as small as possible. Figure 1.3 shows an example with two feasible solutions with the same minimax value of 10, where on the left the fire station is located at node 1 and on the right at node 3. However, when we sort the distances from all nodes to the fire station location from the largest distance to the smallest distance, it becomes obvious that the solution on the right with distances of 10 for node 1, 8 for node 4, 7 for node 2, and 0 for node 3 is better. Indeed, the solution on the right provides the smallest lexicographic vector of distances where the distances are sorted in a nonincreasing order.

This facility location problem is an example where the decision variables (facility locations) are integers. The solution approaches employed for lexicographic minimax problems with continuous decision variables cannot be employed to solve problems with integer variables as the solution of the corresponding minimax problem does not readily provide guidance on how to proceed. The facility location problem with a lexicographic minimax objective

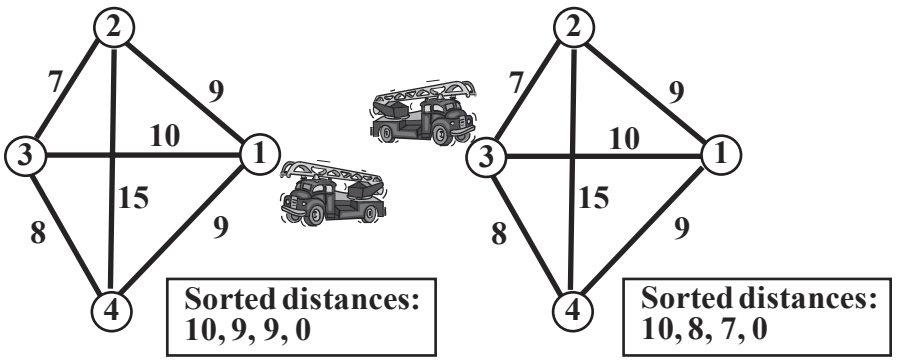

Figure 1.3 Illustration of a facility location problem.

can be solved by an algorithm that repeatedly solves mixed integer programming problems of essentially the same size as that of the minimax formulation of the problem. However, in general, many auxiliary variables and constraints need to be added to solve problems with integer decision variables. Solution methods for problems with integer variables are presented in Chapter 7.

1.3 EXAMPLES AND APPLICATIONS

This section describes a sample of examples taken from diverse application areas. The models presented have a lexicographic minimax (or lexicographic maximin) objective function. Thus, as discussed, the resulting solutions are equitably efficient.

1.3.1 Allocation of High-Tech Components

Typically, high-tech products consist of the assembly of numerous (thousands) components, like integrated circuits, onto a variety (hundreds) of circuit boards. Due to the large number of components and rapid changes in technology, shortages of components are often incurred. In our terminology, the components are the resources and the circuit boards are the activities. The problem of allocating components to circuit boards can then be formulated as Problem L-RESOURCE (with a lexicographic minimax objective; see (1.2.4a)–(1.2.4d)) with the following performance function:

$$f_j(x_j) = \alpha_j \frac{\rho_j - x_j}{\rho_j} \text{ for all } j \in J, \tag{1.3.1}$$

where ρ_j is the demand for activity j, $x_j \le \rho_j$, and α_j is the weight that reflects the relative importance of activity j with respect to other activities. This performance function is the weighted, normalized shortfall from the target demand. If all weights $\alpha_j = 1$, then all activities are equally important. If, for example, $\alpha_1 = 1$ and $\alpha_2 = 2$ (activity 2 is more important than activity 1), then $x_1 = 0.8\rho_1$ and $x_2 = 0.9\rho_2$ have the same weighted normalized shortfall of 0.2. Since $x_j \le \rho_j$ for all $j \in J$, only resources for which $\sum_{j \in J} a_{ij}\rho_j > b_i$ need to be included in the set of resources I as all other resources satisfy demands ρ_j for all $j \in J$. Problem L-RESOURCE is examined in Chapter 3. As will be shown, for linear performance functions, the lexicographic minimax solution is obtained by simply manipulating closed-form expressions; thus, very large problems can be solved in a negligible computing time.

In high-tech manufacturing, substitutions among components are quite common. Effective use of substitutable resources is especially important due to rapidly changing technologies. Hence, extending Problem L-RESOURCE to handling possible substitutions among resources within a subset $SUB \subseteq I$ is important. Let $by(i)$ be the set of resources that can be substituted by

resource i and let $for(i)$ be the set of resources that can substitute for resource i. Let y_{ik} be the amount of resource i used as a substitute for resource k, where y_{ii} is the amount of resource i used directly (not as a substitute). The resource constraints for $i \in SUB$ are then formulated as follows:

$$y_{ii} + \sum_{k \in by(i)} y_{ik} \leq b_i \text{ for all } i \in SUB, \tag{1.3.2a}$$

$$\sum_{j \in J} a_{ij} x_j = y_{ii} + \sum_{k \in for(i)} y_{ki} \text{ for all } i \in SUB. \tag{1.3.2b}$$

Constraints (1.3.2a) ensure that the amount of any resource $i \in SUB$ does not exceed its supply b_i. Constraints (1.3.2b) ensure that the amounts used of resource i and of its substitutes suffice to sustain the selected activity levels for all $j \in J$. If we aggregate all inequalities (1.3.2a) and equalities (1.3.2b) for all $i \in SUB$, we obtain a knapsack resource constraint $\sum_{i \in SUB} \sum_{j \in J} a_{ij} x_j \leq \sum_{i \in SUB} b_i$. This constraint relaxes the restrictions imposed by the allowed substitutions. In other words, it assumes that any of the resources in the set SUB can substitute for any other resource in that set. Various lexicographic minimax resource allocation models with substitutable resources are examined in Chapter 4. The degree of difficulty encountered in solving these problems depends on the structure of the allowed substitutions.

Typically, production planning is executed for a planning horizon that considers multiple periods, for instance, weekly plans for a 6-month period. Extension of the problems described above to a multiperiod setting is examined in Chapter 5.

1.3.2 Throughput in Communication and Computer Networks

Consider a network $G(N, A)$ with a set of nodes N and a set of undirected links A that provides communications between multiple node pairs. The demand between a specified node pair may use flows along multiple paths connecting these nodes (here, the term flow on a specified path means the demand volume routed on the path, or, equivalently, the bandwidth assigned to the demand along that path). Since link capacities are limited, node-pair demands compete on the use of these capacities. The objective is to determine equitable throughputs among all communicating node pairs, where the throughput for a demand between a node pair is the sum of flows of the demand along the multiple paths. We follow what is known as a link-path formulation in networks and use the following notation:

Indices and Sets

e = Index for links in the network.

d = Index for demands between node pairs.

p = Index for paths connecting node pairs in the network.
D = Set of demands in the network; $D = \{1, 2, \ldots, n\}$.
P_d = Set of all paths considered for demand d for all $d \in D$.

Parameters

a_{edp} = 1 if link e is on path p of demand d and 0 otherwise.
b_e = Capacity of link e; $b_e > 0$ for all $e \in A$.

Decision Variables

x_{dp} = Flow of demand d on path p for all $p \in P_d$ and $d \in D$; $\boldsymbol{x}_d = \{x_{dp} : p \in P_d\}$ and $\boldsymbol{x} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n]$.
X_d = Sum of flows (i.e., throughput) of demand d across all paths $p \in P_d$; $X_d = \sum_{p \in P_d} x_{dp}$ for all $d \in D$ and $\boldsymbol{X} = [X_1, X_2, \ldots, X_n]$.

Performance Functions

$w_d(X_d)$ = Performance function of demand d for all $d \in D$.

Figure 1.4 shows a network with four nodes, labeled as 1, 2, 3, 4, and five links, labeled as 1, 2, 3, 4, 5. Consider demand $d = 1$ between nodes 1 and 4. In this example, the links are undirected and the demand is between nodes. The set P_1 of possible paths for demand 1 includes three alternatives as depicted by the dashed curves. Path 1 consists of links 1 and 4, path 2 of link 3, and path 3 of links 2 and 5. The number of feasible paths for each demand can be

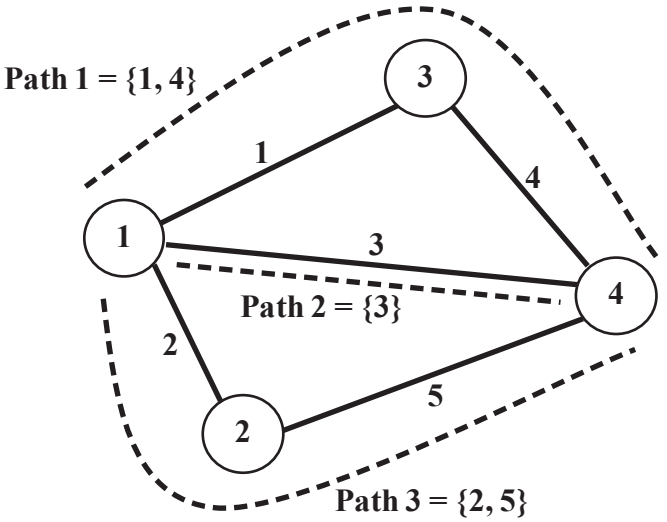

Figure 1.4 A network example with three feasible paths for the demand between nodes 1 and 4.

derived from the network topology, and it may be very large even for moderately sized networks. Quite often, the links are directed, with capacity assigned to a directed link, and the demands are specified from a source node to a destination node. Undirected links are then represented by two directed links.

We now formulate the Flow Allocation Model with Multiple Paths, referred to as Problem L-FAM-MP, that determines lexicographic maximin throughputs for all demands in the network. Thus, we assume the performance function $w_d(X_d) = X_d$ for all $d \in D$. Specifically, the model determines the lexicographic largest vector of throughputs for specified demands where the throughputs are sorted in nondecreasing order. Each demand $d \in D$ is equivalent to an effective activity, while each flow of demand d on path $p \in P_d$ is equivalent to an activity $j \in J$ in Problem L-RESOURCE with a nonseparable objective. The links $e \in A$ are equivalent to resources $i \in I$ in that problem.

PROBLEM L-FAM-MP (lex-maximin objective, $w_d(X_d) = X_d$)

$$\boldsymbol{W}^L = \operatorname*{lexmax}_{\boldsymbol{X}} \{\boldsymbol{w}^{(n)}(\boldsymbol{X}) = [X_{d_1}, X_{d_2}, \ldots, X_{d_n}]\} \tag{1.3.3a}$$

subject to

$$X_{d_1} \le X_{d_2} \le \cdots \le X_{d_n}, \tag{1.3.3b}$$

$$\sum_{d \in D} \sum_{p \in P_d} a_{edp} x_{dp} \le b_e \text{ for all } e \in A, \tag{1.3.3c}$$

$$X_d = \sum_{p \in P_d} x_{dp} \text{ for all } d \in D, \tag{1.3.3d}$$

$$x_{dp} \ge 0 \text{ for all } p \in P_d \text{ and } d \in D. \tag{1.3.3e}$$

The lexicographic maximin objective function is defined by (1.3.3a) and (1.3.3b). Constraints (1.3.3c) ensure that the link capacity constraints are satisfied, while constraints (1.3.3d) define the throughput for each of the demands. Note that Problem L-FAM-MP is the same as Problem L-RESOURCE with a nonseparable objective function where $w_d(X_d) = \sum_{p \in P_d} x_{dp}$ for each $d \in D$ and the throughput X_d of demand d is achieved by flows along one or more paths $p \in P_d$. Indeed, let $\omega_d(x_{d1}, x_{d2}, \ldots, x_{d|P_d|}) = \omega_d(\boldsymbol{x}_d) = \sum_{p \in P_d} x_{dp}$ for all $d \in D$ ($|P_d|$ denotes the number of paths in P_d). Hence, we can remove variable X_d and obtain an equivalent formulation.

PROBLEM L-FAM-MP (lex-maximin objective, $\omega_d(\boldsymbol{x}_d) = \sum_{p \in P_d} \boldsymbol{x}_{dp}$)

$$\boldsymbol{W}^L = \operatorname*{lexmax}_{\boldsymbol{x}} \{\boldsymbol{\omega}^{(n)}(\boldsymbol{x}) = [\omega_{d_1}(\boldsymbol{x}_{d_1}), \omega_{d_2}(\boldsymbol{x}_{d_2}), \ldots, \omega_{d_n}(\boldsymbol{x}_{d_n})]\} \tag{1.3.4a}$$

subject to

$$\omega_{d_1}(\boldsymbol{x}_{d_1}) \le \omega_{d_2}(\boldsymbol{x}_{d_2}) \le \cdots \le \omega_{d_n}(\boldsymbol{x}_{d_n}), \tag{1.3.4b}$$

$$\sum_{d \in D} \sum_{p \in P_d} a_{edp} x_{dp} \le b_e \text{ for all } e \in A, \tag{1.3.4c}$$

$$x_{dp} \ge 0 \text{ for all } p \in P_d \text{ and } d \in D. \tag{1.3.4d}$$

In this equivalent formulation, it is obvious that the objective function of Problem L-FAM-MP is nonseparable.

Algorithms for solving such problems will be discussed in Sections 3.4 and 6.2. When the routing of each of the demands is limited to a single fixed path, the formulation reduces to a special case of Problem L-RESOURCE, where the links are the resources and the demands are the activities. This problem is discussed in Section 6.1.

1.3.3 Point-to-Point Throughput Estimation in Networks

Telecommunication and transportation network design methods are employed to plan the expansion of critical link capacities for a long planning horizon and to change routing patterns over relatively short time frames. Typically, these methods require as input estimates of throughputs that need to be supported between all node pairs, which are not readily available. Numerous variants of point-to-point throughput estimation problems in networks have been published (the first paper was published by Kruithof in 1937), and many solution approaches have been proposed. Here, we present a model for a network that carries multiple service types with different characteristics, for example, different types of data. The information available includes total carried load (average carried traffic per time unit) on each link, and originating load per service at each node and terminating load per service at each node. Viewing these collected load information as "resources" that should be allocated equitably among all demands, a throughput estimation problem with a lexicographic minimax objective can be formulated. Let $G(N, A)$ be a network with a set of nodes N and a set of directed links A. We use the following notation:

Indices and Sets

e = Index for directed links in the network.

i, j = Indices for nodes in the network.

s = Index for services.

d = Index for demands. A demand d uniquely implies a triplet (s, i, j), that is, the service s, its originating node i, and its terminating node j.

D = Set of demands in the network; $D = \{1, 2, \ldots, n\}$.

S = Set of services.

I_s = Set of nodes where service s is originated.

J_s = Set of nodes where service s is terminated.

Parameters

β_{ed} = Fraction of throughput of demand d that is carried on link e. Depending on the application, this input is available from historical data or from routing protocols.

γ_{isd} = 1 if demand d is for service s and originates at node i; otherwise, $\gamma_{isd} = 0$.

δ_{jsd} = 1 if demand d is for service s and terminates at node j; otherwise, $\delta_{jsd} = 0$.

L_e = Load measured on link e for all $e \in A$.

O_{is} = Load measured for service s, originated at node i for all $i \in I_s$ and $s \in S$.

T_{js} = Load measured for service s, terminated at node j for all $j \in J_s$ and $s \in S$.

ρ_d = Rough throughput estimate for demand d for all $d \in D$. These estimates can be based on demographic information and simple models, such as gravity models or Kruithof's method. $\rho_d > 0$ for all $d \in D$.

Decision Variables

x_d = Throughput estimate for demand d for all $d \in D$. These estimates are the output of the model. They depend on the rough estimates ρ_d, the routing fractions β_{ed}, and on all load measurements L_e, O_{is}, and T_{js}.

Performance Functions

$f_d(x_d)$ = Deviation of the throughput estimate of demand d from its rough estimate; $f_d(x_d) = (\rho_d - x_d)/\rho_d$ for all $d \in D$.

Since the demand throughput estimates depend on load measurements, the model can be employed for different time periods, thus providing estimates that are consistent with load measurements for each particular time. The Throughput Estimation Problem, referred to as Problem L-THPUTEST, is formulated as follows.

PROBLEM L-THPUTEST (lex-minimax objective)

$$\boldsymbol{V}^L = \operatorname*{lexmin}_{\boldsymbol{x}} \{\boldsymbol{f}^{(n)}(\boldsymbol{x}) = [f_{d_1}(x_{d_1}), f_{d_2}(x_{d_2}), \ldots, f_{d_n}(x_{d_n})]\} \qquad (1.3.5a)$$

subject to

$$f_{d_1}(x_{d_1}) \geq f_{d_2}(x_{d_2}) \geq \cdots \geq f_{d_n}(x_{d_n}), \tag{1.3.5b}$$

$$\sum_{d \in D} \beta_{ed} x_d \leq L_e \text{ for all } e \in A, \tag{1.3.5c}$$

$$\sum_{d \in D} \gamma_{isd} x_d \leq O_{is} \text{ for all } i \in I_s \text{ and } s \in S, \tag{1.3.5d}$$

$$\sum_{d \in D} \delta_{jsd} x_d \leq T_{js} \text{ for all } j \in J_s \text{ and } s \in S, \tag{1.3.5e}$$

$$x_d \geq 0 \text{ for all } d \in D, \tag{1.3.5f}$$

where $f_d(x_d) = (\rho_d - x_d)/\rho_d$ for all $d \in D$. The lexicographic minimax objective function is defined by (1.3.5a) and (1.3.5b). Constraints (1.3.5c) ensure that, for each link, the total flow assumed to be routed through a link does not exceed the link load measurement. Constraints (1.3.5d) ensure that, for each originating node of service s, the total load for service s originated at that node and routed to all destinations does not exceed the corresponding load measurement. Likewise, constraints (1.3.5e) ensure the same for each terminating node of each service. The formulation of Problem L-THPUTEST is in the same format as the formulation of Problem L-RESOURCE and is solved by the same algorithms. Note that the link and node load measurements do not account for lost traffic due to congestion, failures, and so on. Hence, in order to estimate demand volumes between node pairs, some adjustments are needed to account for lost traffic.

The model attempts to compute equitable throughput estimates using the static estimates ρ_d for all $d \in D$ as benchmarks. In addition, the lexicographic minimax objective attempts to provide a solution that is consistent with all load measurements by attempting to satisfy all resource constraints at equalities. Computational results for Problem L-THPUTEST are shown in Section 3.3. Indeed, large problems can be solved with negligible computational effort.

A similar lexicographic minimax model can be employed for estimating spatial loads in cellular wireless networks. Consider an area of 40 km by 40 km, partitioned into 40,000 "bins" of size 200 m by 200 m, and served by 200 base transceiver stations. Using load measurements at the base stations (200 measurements) and propagation models that compute probabilities of assigning each bin to each of the base stations, estimates of loads originating at each of the 40,000 bins can be computed. Since, typically, the load measurements at the base stations are aggregated over all services, the formulation of Problem L-THPUTEST for this application considers a single service without nodal measurements.

1.3.4 Bandwidth Allocation for Content Distribution

Content distribution over networks has become increasingly popular. Primary application areas include on-demand home entertainment, remote learning

and training, video conferencing, and news on demand. Service providers must provide significant bandwidth resources in order to provide adequate service, which requires large capital investments. Consider, for example, near video-on-demand (VOD) applications where a server broadcasts a copy of a popular movie every 5 minutes meaning that a customer may wait, on average, 2.5 minutes. Thus, if the length of the movie is 100 minutes, the network must carry 20 copies of the movie simultaneously. In other delivery technologies that provide almost instantaneous VOD delivery, bandwidth allocation to any program can be controlled by changing the video quality provided to users. The quality expected by customers may depend on the application. For example, broadcasting of movies requires better video quality than multicasting video conferences.

Figure 1.5 shows a tree network with a server at its root (node 1) that broadcasts programs 1, 2, 3, and 4. Each node has requests for a subset of these programs. For example, node 2 requests programs 1 and 4 and node 3 requests program 3. The programs are routed along the tree where each of the links carries at most a single copy of the same program. The boxes adjacent to the links indicate the link index and which programs are carried on each of the links. For example, link 1 (between nodes 1 and 2) carries programs 1, 2, 3, and 4, and link 3 (between nodes 2 and 4) carries programs 1, 3, and 4. Each link has a limited bandwidth capacity that is allocated among the programs carried on the link. Suppose program 3 is allocated 100 Mbps on link 1. The bandwidth allocated to program 3 on links 3 or 4 can be less than 100 Mbps, but cannot exceed 100 Mbps. These constraints are referred to as *treelike*

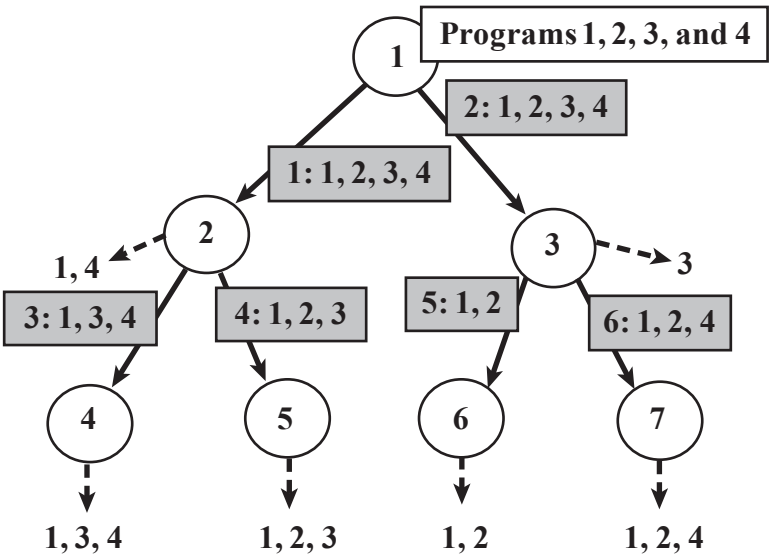

Figure 1.5 A tree network for content distribution.

ordering constraints. As a result, links that are farther away from the server may end up with excess capacities. Each of the requested programs is associated with a performance function that represents service satisfaction as a function of the incoming bandwidth of that program into the node requesting the program. The objective is to allocate the bandwidths along all links so that the service provided to all requested programs at the different nodes is equitable.

Let $G(N, A)$ be a tree network with a set of nodes N and a set of directed links A. We use the following notation:

Indices and Sets

i, j = Indices for nodes.

e, m = Indices for directed links.

p = Index for programs.

SUC_e = Set of immediate successor links of link e; for example, in Figure 1.5, $SUC_2 = \{5, 6\}$.

P = Set of programs broadcasted from the server at root node 1.

D_j = Set of programs requested at node j; $D_j \subseteq P$.

Parameters

b_e = Bandwidth capacity of link e; $b_e > 0$ for all $e \in A$.

l_p = Lower bound ($l_p \geq 0$) for bandwidth required by program p for all $p \in P$.

More Sets (Derived from the Set Dj and the Tree Topology)

LP_e = Set of programs that are carried on link e for all $e \in A$. These are shown in Figure 1.5 for each link.

NP = $\{(j, p) \mid p \in D_j, j \in N\}$; that is, doubleton $(j, p) \in NP$ if program p is requested at node j. Let n be the number of doubletons in NP.

LP = $\{(e, p) \mid p \in LP_e, e \in A\}$; that is, doubleton $(e, p) \in LP$ if program p is carried on link e.

Decision Variables

x_{ep} = Bandwidth allocated on link e to program p for all $(e, p) \in LP$; $\boldsymbol{x} = \{x_{ep}: (e, p) \in LP\}$.

Performance Functions

$w_{jp}(x_{ep})$ = Performance function associated with program p requested at node j, representing service quality satisfaction, for all $(j, p) \in NP$. The performance functions are strictly increasing with x_{ep}, where the incoming link e into node j is unique.

We now formulate the Bandwidth Allocation Model with a Single Server, referred to as Problem L-BAM-SS, as a lexicographic maximin optimization problem.

PROBLEM L-BAM-SS (lex-maximin objective)

$$\boldsymbol{W}^L = \operatorname*{lexmax}_{\boldsymbol{x}} \{\boldsymbol{w}^{(n)}(x) = [w_{j_1p_1}(x_{e_1p_1}), w_{j_2p_2}(x_{e_2p_2}), \ldots, w_{j_np_n}(x_{e_np_n})]\} \tag{1.3.6a}$$

subject to

$$w_{j_1p_1}(x_{e_1p_1}) \le w_{j_2p_2}(x_{e_2p_2}) \le \cdots \le w_{j_np_n}(x_{e_np_n}), \tag{1.3.6b}$$

$$\sum_{p \in LP_e} x_{ep} \le b_e \text{ for all } e \in A, \tag{1.3.6c}$$

$$x_{ep} \ge x_{mp} \text{ for all } m \in SUC_e \text{ and } p \in LP_m, \text{ for all } (e, p) \in LP, \tag{1.3.6d}$$

$$x_{ep} \ge l_p \text{ for all } (e, p) \in LP. \tag{1.3.6e}$$

To clarify the notation, consider $w_{j_1p_1}(x_{e_1p_1})$. This is the performance function at node j_1 for program p_1 as a function of the bandwidth allocated to program p_1 on link e_1, where link e_1 is the incoming link into node j_1. For example, in Figure 1.5, $w_{41}(x_{31})$ is the performance function at node 4 for program 1 as a function of the bandwidth allocated to program 1 on link 3. We assume that $\sum_{p \in LP_e} l_p \le b_e$ for all $e \in A$ so that a feasible solution to Problem L-BAM-SS exists. The lexicographic maximin objective function is defined by (1.3.6a) and (1.3.6b). Constraints (1.3.6c) enforce the bandwidth capacity constraint on each of the links. Constraints (1.3.6d) enforce the treelike ordering constraints. Under these constraints, for each program, the allocated bandwidth along the links may decrease, but not increase, when moving farther away from the root node. Due to these constraints, bandwidth allocated to a specific program on a link close to the root node may limit the bandwidth that can be allocated to that program on farther away links even if these links have excess capacity. Constraints (1.3.6e) enforce lower bounds on bandwidth allocation for each program on each link that carries the program. Variants of such bandwidth allocation problems for content distribution purposes are examined in Sections 6.3 and 6.4.

1.3.5 Location of Emergency Facilities

The problem of locating emergency facilities, such as fire or police stations, at a subset of nodes of a network was briefly discussed in Section 2.1. Figure 1.3 demonstrated the value of seeking a lexicographic minimax solution rather than being satisfied with an arbitrary minimax solution, where only the largest distance between a node and its assigned facility is minimized. Let $G(N, A)$

be a network with a set of nodes N and a set of undirected links A. We use the following notation:

Indices

i, j = Indices for nodes.

Parameters

d_{ij} = Length of shortest path between nodes i and j for all $i, j \in N$ ($d_{ii} = 0$).

K = Number of facilities that can be installed; $K < n$ (where n is the number of nodes in N).

Decision Variables

x_i = 1 if a facility is located at node i, and $x_i = 0$ otherwise; $\boldsymbol{x} = \{x_1, x_2, \ldots, x_n\}$.

y_{ij} = 1 if a facility at node i is assigned to serve node j, and $y_{ij} = 0$ otherwise, for all $i, j \in N$. Note that $\boldsymbol{x}$ readily implies the assignment of nodes to facilities (j is assigned to the closest facility); $\boldsymbol{y} = \{y_{ij}: i, j \in N\}$.

Performance Functions

$f_j(\boldsymbol{x})$ = The distance of node j to its assigned facility (the closest one) as a function of all facility locations, as specified by $\boldsymbol{x}$, for all $j \in N$.

We now formulate the Facility Location Model with a lexicographic minimax objective, referred to as Problem L-FLM. Note that this problem is quite different from those described before. Specifically, the problem has 0–1 decision variables. The performance functions are not separable since, for each j, $f_j(\boldsymbol{x})$ depends on the vector $\boldsymbol{x}$ and not just on x_j.

PROBLEM L-FLM (lex-minimax objective)

$$\boldsymbol{V}^L = \underset{\boldsymbol{x},\boldsymbol{y}}{\text{lexmin}}\{\boldsymbol{f}^{(n)}(\boldsymbol{x}) = [f_{j_1}(\boldsymbol{x}), f_{j_2}(\boldsymbol{x}), \ldots, f_{j_n}(\boldsymbol{x})]\} \tag{1.3.7a}$$

subject to

$$f_{j_1}(\boldsymbol{x}) \geq f_{j_2}(\boldsymbol{x}) \geq \cdots \geq f_{j_n}(\boldsymbol{x}), \tag{1.3.7b}$$

$$f_j(\boldsymbol{x}) = \sum_{i \in N} d_{ij} y_{ij} \text{ for all } j \in N, \tag{1.3.7c}$$

$$\sum_{i \in N} x_i = K, \tag{1.3.7d}$$

$$y_{ij} \leq x_i \text{ for all } i, j \in N, \tag{1.3.7e}$$

$$\sum_{i \in N} y_{ij} = 1 \text{ for all } j \in N, \tag{1.3.7f}$$

$$x_i = 0 \text{ or } 1 \text{ for all } i \in N, \tag{1.3.7g}$$

$$y_{ij} = 0 \text{ or } 1 \text{ for all } i, j \in N. \tag{1.3.7h}$$

Constraints (1.3.7c) define the performance functions. For example, if a facility at node 1 is assigned to serve node 5, then, by (1.3.7f) $y_{15} = 1$ while $y_{i5} = 0$ for all $i \neq 5$, which implies by (1.3.7c) that the performance function value of node 5 is equal to the distance d_{15} between nodes 1 and 5. Constraint (1.3.7d) limits the number of facilities (the resources) to K. Constraints (1.3.7e) ensure that nodes will not be assigned to be served by another node that does not have a facility. Constraints (1.3.7f) ensure that all nodes will be assigned to a facility. Constraints (1.3.7g) and (1.3.7h) limit the decision variable values to 0 or 1. An algorithm for solving this problem is described in Section 7.2.

1.3.6 Other Applications

The examples described in this section hopefully provided the reader with some insight regarding the broad applicability of equitable resource allocation models with a lexicographic minimax (or maximin) objective function to numerous diverse areas. Allocation of strategic resources for military applications is another obvious area, for example, the distribution of spare parts to various weapon systems among different locations, and the assignment of different types of aircrafts for the execution of multiple missions. Equitable scheduling issues faced by the U.S. Federal Aviation Administration (FAA) are critically important. Collaborative air traffic flow management addresses issues such as allocation of takeoff and landing slots and allocation of airspace. Energy resources and water resources are often in short supply and should be allocated equitably and efficiently among users. Other examples include allocation of inspection effort among numerous components of large systems and allocation of effort for software reliability. In many countries, significant attention has been devoted to improving health services under limited resources. Many health care decisions would benefit from applying equitable resource allocation models, including allocating beds among competing hospital departments, allocating expensive equipment among hospitals in a region, and providing incentives to have the right mix of specialist physicians.

Equitable resource allocation models are obviously important for service-related applications as society expects that services will be provided fairly. Nevertheless, as our examples demonstrate, such models play an important role in numerous, diverse application areas. We believe that the message is clear: There are endless challenges and opportunities to apply variations of such resource allocation models in order to solve important problems that would benefit the private and public sectors, and society at large.

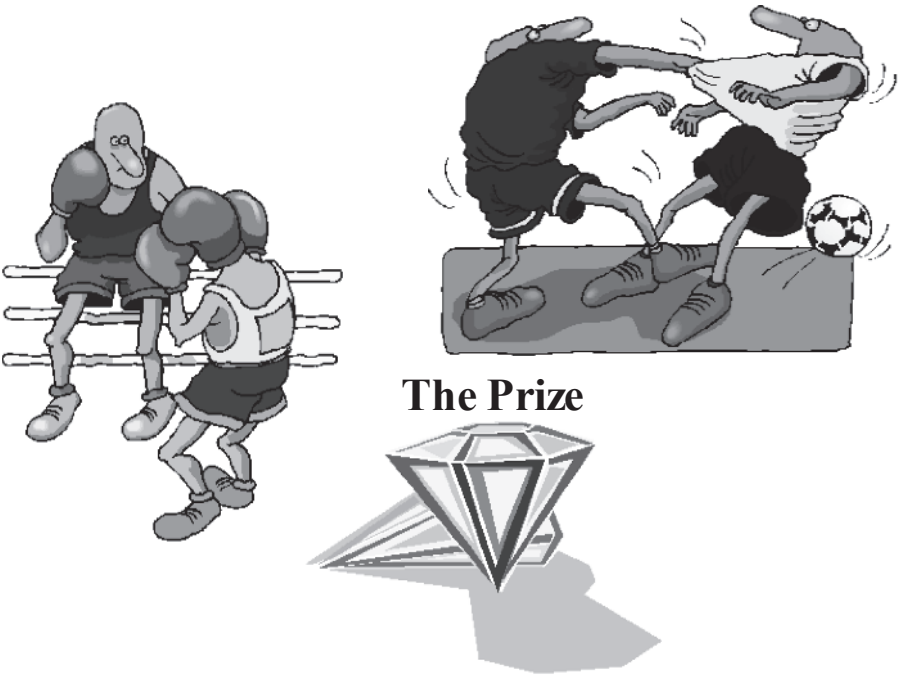

Figure 1.6 Winner takes all.

1.4 RELATED FAIRNESS CRITERIA

The issue of fairness has received considerable attention in the literature. In fact, it has been practiced since ancient times under the philosophy of "winner takes all" as demonstrated mildly in Figure 1.6.

This book focuses on resource allocation models with a *lexicographic minimax* (or *lexicographic maximin*) objective function. The merit of a lexicographic minimax (or lexicographic maximin) objective function has been explained in this chapter through intuitive arguments without resorting to axiomatic analysis. As already discussed, a well-known notion is that of a *pareto-optimal* (*efficient*) solution, where no performance function value of any activity can be feasibly improved without worsening that of another activity. It is easy to see that a minimax (or maximin) solution is not necessarily pareto-optimal, whereas a lexicographic minimax (or lexicographic maximin) solution is. A lexicographic minimax (maximin) solution selects a particularly attractive pareto-optimal solution that is also equitable. Hence, it is often referred to as an *equitably efficient* solution. In some sense, the lexicographic minimax (maximin) solution can also be referred to as the "most equitable solution" since it provides the best feasible performance function value for activities with the worst performance function value, followed by the best

feasible performance function value for activities with the second worst performance function value without degrading the worst value, and so forth.

A minimax (maximin) objective is partially equitable according to the Rawlsian theory of justice, which implies that the greater social and economic advantages of society should not be at the expense of the least fortunate. However, a minimax (maximin) solution, although fair for the worst-off activity, does not give any guidance how to select an equitable solution across all activities. This gap is filled by the lexicographic minimax (maximin) objective.

Consider performance vector $\boldsymbol{f}(\boldsymbol{x}) = [f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_n(\boldsymbol{x})]$ for any feasible $\boldsymbol{x}$. The vector $\boldsymbol{f}^{(n)}(\boldsymbol{x}) = [f_{j_1}(\boldsymbol{x}), f_{j_2}(\boldsymbol{x}), \ldots, f_{j_n}(\boldsymbol{x})]$ is generated from $\boldsymbol{f}(\boldsymbol{x})$ by sorting the performance functions in nonincreasing order (the discussion below is for lexicographic minimax optimization). The lexicographic minimax vector satisfies various preference relations, which are perceived as desired properties of equitable solutions. Let z_j be the outcome of performance function j for a given decision vector $\boldsymbol{x}$; that is, $z_j = f_j(\boldsymbol{x})$ for $j = 1, 2, \ldots, n$, and in vector notation, $\boldsymbol{z} = \boldsymbol{f}(\boldsymbol{x})$. The discussion below is limited to the criteria (performance function outcomes) space and let the set of feasible outcome vectors be $\boldsymbol{Z}$. We use well-known terminology of preference relations as follows: Vector of outcomes $\boldsymbol{z}^1$ is weakly preferred over $\boldsymbol{z}^2$ ($\boldsymbol{z}^1 \preceq \boldsymbol{z}^2$), vector $\boldsymbol{z}^1$ is strictly preferred over $\boldsymbol{z}^2$ ($\boldsymbol{z}^1 \prec \boldsymbol{z}^2$), and vectors $\boldsymbol{z}^1$ and $\boldsymbol{z}^2$ are equally preferred ($\boldsymbol{z}^1 \cong \boldsymbol{z}^2$). Equitable solutions should satisfy the following properties:

(a) *Completeness:* Either $\boldsymbol{z}^1 \preceq \boldsymbol{z}^2$ or $\boldsymbol{z}^2 \preceq \boldsymbol{z}^1$ for any $\boldsymbol{z}^1, \boldsymbol{z}^2 \in \boldsymbol{Z}$.

(b) *Transitivity:* If $\boldsymbol{z}^1 \preceq \boldsymbol{z}^2$ and $\boldsymbol{z}^2 \preceq \boldsymbol{z}^3$, then $\boldsymbol{z}^1 \preceq \boldsymbol{z}^3$ for any $\boldsymbol{z}^1, \boldsymbol{z}^2, \boldsymbol{z}^3 \in \boldsymbol{Z}$.

(c) *Strictly Monotonic:* $\boldsymbol{z} - \varepsilon \boldsymbol{e}_j \prec \boldsymbol{z}$ for any $(\boldsymbol{z} - \varepsilon \boldsymbol{e}_j), \boldsymbol{z} \in \boldsymbol{Z}$ and $j = 1, 2, \ldots, n$, where $\boldsymbol{e}_j$ is the jth unit vector and ε is an arbitrarily small positive constant.

(d) *Scale Invariance:* If $\boldsymbol{z}^1 \preceq \boldsymbol{z}^2$, then $c\boldsymbol{z}^1 \preceq c\boldsymbol{z}^2$ for any $\boldsymbol{z}^1, \boldsymbol{z}^2 \in \boldsymbol{Z}$ and scalar $c > 0$.

(e) *Anonymity (Impartiality):* $\boldsymbol{z}^1 \cong \boldsymbol{z}^2$ if $\boldsymbol{z}^2$ is a permutation of the elements of $\boldsymbol{z}^1$ for any $\boldsymbol{z}^1, \boldsymbol{z}^2 \in \boldsymbol{Z}$.

(f) *Principle of Transferability:* If $z_{j'} > z_{j''}$, then $\boldsymbol{z} - \varepsilon \boldsymbol{e}_{j'} + \varepsilon \boldsymbol{e}_{j''} \prec \boldsymbol{z}$ for any $0 < \varepsilon < z_{j'} - z_{j''}$ and $(\boldsymbol{z} - \varepsilon \boldsymbol{e}_{j'} + \varepsilon \boldsymbol{e}_{j''}), \boldsymbol{z} \in \boldsymbol{Z}$.

The most interesting properties for our purposes are (e) and (f). Indeed, the vector $\boldsymbol{f}^{(n)}(\boldsymbol{x})$ is the same for any permutation of $f_j(\boldsymbol{x})$ for $j = 1, 2, \ldots, n$, implying that an equitable solution is impartial with respect to the specific values of j. For instance, if $\boldsymbol{z}^1 = [f_1^1(\boldsymbol{x}) = 10, f_2^1(\boldsymbol{x}) = 5]$ and $\boldsymbol{z}^2 = [f_1^2(\boldsymbol{x}) = 5, f_2^1(\boldsymbol{x}) = 10]$, then $\boldsymbol{z}^1 \cong \boldsymbol{z}^2$. The principle of transferability states that improving a worse-off outcome at the expense of a better-off one is a preferred solution, and the preference relation is maintained as long as the change is within the specified interval. For instance, suppose that $\boldsymbol{z}^1 = [z_1^1 = 10, z_2^1 = 5, z_3^1 = 2]$. Then,

$\boldsymbol{z}^2 = [z_1^2 = 10 - 7 = 3, z_2^2 = 5, z_3^2 = 2 + 7 = 9] \prec \boldsymbol{z}^1$. The lexicographic minimax objective satisfies the properties of equitable solutions as described above. These preference relations can readily be modified for a lexicographic maximin objective.

A notion of fairness, referred to as *max-min fairness*, is often used in network flow problems. Consider, for example, Problem L-FAM-MP in Section 1.3 that considers throughput in communication and computer networks. Suppose $\boldsymbol{x}^0$ (and $\boldsymbol{X}^0$) satisfy constraints (1.3.3c)–(1.3.3e). Then, $\boldsymbol{x}^0$ (and $\boldsymbol{X}^0$) is a max-min fair solution if and only if for any other feasible solution $\boldsymbol{x}$ (and $\boldsymbol{X}$), $w_{d_2}(X_{d_2}) > w_{d_2}(X^0_{d_2})$ implies the existence of demand d_1 such that $w_{d_1}(X_{d_1}) < w_{d_1}(X^0_{d_1}) \leq w_{d_2}(X^0_{d_2})$. In other words, a max-min fair solution implies that no performance function value can be feasibly increased without decreasing the performance function value of some other activity whose performance function value is already at least as small. Thus, the max-min fair and lexicographic maximin solutions are the same for problem L-FAM-MP. In general, the max-min fair solution (as defined above) and the lexicographic maximin solution are equivalent as long as the feasible region is convex and compact, and the performance functions are continuous and strictly increasing (or concave). These conditions will be satisfied for all models in Chapters 3–6. However, for problems with discrete or integer decision variables, discussed in Chapter 7, max-min fair solutions and lexicographic maximin solutions are not equivalent. Consider the following simple example with two activities and a single resource constraint: $w_1(x_1) = x_1$, $w_2(x_2) = x_2$ subject to $x_1 + x_2 = 10$, $x_1 \geq 0$, and $x_2 \geq 0$. Both the max-min fair solution and the lexicographic maximin solution are $x_1 = x_2 = 5$. Now, suppose that the only feasible solutions are either $x_1 = 3$ and $x_2 = 7$, or $x_1 = 8$ and $x_2 = 2$. It is easy to see that whereas the first solution is the lexicographic maximin solution, neither solution is a max-min fair solution.

It is important to note that significant work has been published on a variety of fairness issues. After all, seeking fairness in societal issues has captured the energy of people for centuries with only partial success. For an in-depth discussion on fairness, the reader will be referred to representative references. The discussion below highlights only some related concepts.

A well-known fairness notion that has been motivated by communication network applications is that of proportional fairness. Consider a special case of the network example in Section 1.3.2, where each demand is routed on a specified single path. Suppose that the lexicographic maximin objective is replaced by $\max_{\boldsymbol{x}} \sum_{d \in D} \log x_d$. It can be shown that the optimal solution $\boldsymbol{x}^{pf}$ satisfies

$$\sum_{d \in D} [(x_d - x_d^{pf}) / x_d^{pf}] \leq 0, \tag{1.4.1}$$

where x_d, $d \in D$, is any feasible solution. Solution $\boldsymbol{x}^{pf}$ is called proportionally fair since the aggregate of proportional changes with respect to any other

feasible solution is zero or negative. The proportional fairness concept has been extended by adding positive weights to each term in the objective function. It has also been applied to more complicated models like the network example with multiple routes. Of course, we can use other strictly concave performance functions instead of the logarithmic function. A logarithmic function is convenient as it eliminates the case of zero allocation to any of the demands ($\log(0) = -\infty$). Whereas the lexicographic maximin objective gives priority to fairness over resource utilization, proportional fairness better utilizes the resources at the expense of fairness.

A family of performance functions that includes proportional fairness and lexicographic maximin objective (with $w_d(x_d) = x_d$ for all $d \in D$) is known as the α-fair utility function:

$$w_\alpha(x_d) = \frac{x_d^{1-\alpha}}{1-\alpha} \text{ for } \alpha \geq 0, \alpha \neq 1, \text{ and } d \in D, \tag{1.4.2a}$$

$$w_\alpha(x_d) = \log(x_d) \text{ for } \alpha = 1 \text{ and } d \in D, \tag{1.4.2b}$$

where the objective is to maximize the sum of the utility functions over all demands. This utility function captures a whole family of criteria by selecting different values for α. When $\alpha = 1$, the optimal solution is proportionally fair and has been shown to be effective for rate control in communication networks.

When $\alpha = 0$, the utility function is linear. When $\alpha \to \infty$, the function approximates the lexicographic maximin objective. It is quite difficult, though, to understand the meaning of some arbitrary value of α. Nevertheless, for sufficiently large α, maximizing the sum of utility functions (1.4.2a) may serve as an approximation to the lexicographic maximin objective.

The topic of multiobjective optimization, where a collection of objective functions are optimized, has been thoroughly studied. For instance, lexicographic optimization methods order the criteria according to specified importance and then optimize first the most important criterion, followed by the second most important criterion without degrading the value achieved for the first criterion, and so forth. These methods do not necessarily provide equitable solutions; for instance, they do not satisfy impartiality since they prioritize in advance the order in which the criteria are optimized. Lexicographic optimization methods, with a predetermined order in which criteria are optimized, should not be confused with lexicographic minimax (or maximin) methods. Other multiobjective methods use an objective function that is a weighted composite of the individual criteria. Weighted proportional fairness ($\max_x \sum_{d \in D} \alpha_d \log x_d$ with $\alpha_d > 0$) is an example for such an objective function. Again, such methods do not necessarily provide equitable solutions. A modification of these methods, referred to as the ordered weighted average method, assigns the largest weight to the worst performance function value, the second largest weight to the second worst performance function value, and so forth.

Indeed, under certain conditions, the ordered weighted average solution approximates the lexicographic minimax (or maximin) solution.

1.5 OUTLINE OF THE BOOK

1.5.1 Chapter 2: Nonlinear Resource Allocation

Chapter 2 covers initially the well-studied problem of maximizing a separable objective function subject to a single linear resource constraint. The decision variables are assumed to be continuous and represent activity levels. The objective function maximizes the sum of concave performance functions (or minimizes the sum of convex functions), where each of these functions depends on the level selected for one activity. This problem has proven to be valuable in many applications. Since the marginal return of any specific activity decreases with the amount of resource allocated to that activity, resources are allocated somewhat "fairly" among the activities. For example, as discussed before, proportional fairness is achieved when the sum of logarithmic performance functions is maximized. For certain classes of performance functions, algorithms that compute optimal solutions by manipulating closed-form expressions are presented. For other functions, we present an algorithm that employs function evaluations and a numerical search.

The algorithms are then extended to maximizing the sum of concave functions subject to a resource constraint composed of the sum of convex resource-usage functions. Again, for certain classes of performance functions, the algorithms compute optimal solutions by manipulating closed-form expressions, while for other functions, a numerical search is employed. As will be shown, large problems can be solved with a small computational effort.

The final topic considers an example taken from allocating different promotional activities across multiple regions. The problem is formulated with a nonseparable objective function that maximizes the sum of performance functions subject to multiple resource constraints, and is solved by repeatedly solving problems with a separable objective function and a single resource constraint.

Although the models do not provide equitable solutions, some of the algorithms are quite similar to the minimax algorithms presented in Chapter 3. Hence, this chapter serves as a worthwhile lead into equitable resource allocation models.

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

Chapters 3–6 present a large variety of resource allocation models with a lexicographic minimax or maximin objective function and continuous decision variables.

Chapter 3 first covers the basic equitable resource allocation problem, formulated as Problem L-RESOURCE. The problem considers a lexicographic minimax separable objective function subject to multiple knapsack resource constraints, and lower and upper bounds on activity levels. The lexicographic minimax algorithm repeatedly solves minimax problems. After each such problem is solved, some activity levels are fixed at their lexicographic minimax value and are removed from subsequent minimax problems. This lexicographic minimax (or maximin) optimization framework is repeated throughout Chapters 3–6. The key difference in various problems lies in the algorithms for solving the minimax problems. For certain classes of performance functions, the algorithms for Problem L-RESOURCE are extremely efficient as they only require manipulating closed-form expressions. For more general performance functions, the algorithms require more intensive computations. Similarities between the algorithms presented in Chapter 2 and the minimax algorithms presented here are quite striking, especially when the minimax problem has only one resource constraint.

This chapter also discusses algorithms for a nonseparable objective function, where each performance function depends on a linear combination of multiple activity levels, referred to as the effective activity level. These problems are significantly more difficult than those with a separable objective function, even with linear performance functions. We present an algorithm for linear performance functions that repeatedly solves minimax problems as linear programming problems and identifies saturated effective activities after each such solution is computed. Saturated effective activities are excluded from the objective function of subsequent minimax problems, while their saturation level is protected through newly added constraints.

The work described in this chapter was initially motivated by the allocation of critical components for the manufacturing of high-tech products. It is, however, applicable to many other areas, including communication networks, transportation, logistics, and water resources. For instance, the basic problem with a separable objective function is directly applicable to determining the lexicographic maximin throughputs in communication networks with a single fixed path for each demand. The model with a nonseparable objective function is directly applicable to determining the lexicographic maximin throughputs in communication networks with multiple feasible paths for each demand.

1.5.3 Chapter 4: Equitable Resource Allocation with Substitutable Resources

Chapter 4 extends Problem L-RESOURCE of Chapter 3 to models where subsets of resources are substitutable. 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 considers different structures of substitutions and presents efficient algorithms for these cases. The first structure considers transitive substitutable resources represented by networks with a tree topology. Transitivity among substitutable resources means that if resource i_1 can substitute for resource i_2 and resource i_2 can substitute for resource i_3, then resource i_1 can also substitute for resource i_3. Representation by a tree topology implies that each resource, except for the root node of the tree, can be substituted directly (not by transitivity) by exactly one other resource. The second structure considers transitive resources represented by more general acyclic networks. Here, a resource may be substituted directly by multiple resources, adding significant flexibility to resource allocation decisions. The third structure considers nontransitive substitutable resources. The last structure considers activity-dependent substitutable resources, for example, where resource i_1 may substitute for resource i_2 when used for activity j_1, but not when used for activity j_2.

Again, as will be shown, the lexicographic minimax algorithms for these problems repeatedly solve minimax problems. However, the algorithms for the minimax problems are more complicated and consist of two major components. The first component repeatedly solves relaxed minimax problems by aggregating substitutable resource constraints, resulting in problems that are in the format of Problem L-RESOURCE. The second component checks whether the solution to the relaxed problem is feasible or not for the original minimax problem. If not feasible, it identifies subsets of resources with sufficient supplies that can be deleted from subsequent relaxed problems. The methods for identifying such subsets depend on the structures of the substitutions, and vary from employing simple backtracking schemes to solving max-flow network problems. Other solution approaches for the minimax problem are also presented.

1.5.4 Chapter 5: Multiperiod Equitable Resource Allocation

Chapter 5 extends the models of Chapter 3 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 integrated circuits used in high-tech products, water reserves, oil reserves, and so forth. The multiperiod model for storable resources extends the formulation of Problem L-RESOURCE by adding a sequence of ordering constraints for each activity that restrict selected cumulative activity levels for each activity to be nondecreasing over time. Several minimax algorithms are presented for this problem. The first algorithm is based on a numerical search. It can handle quite general performance functions, but its computational effort is pseudo-polynomial as it depends on the desired accuracy. The second algorithm is particularly efficient for classes of

performance functions that solve Problem L-RESOURCE of Chapter 3 by manipulating closed-form expressions, since it repeatedly solves such problems. The third algorithm is tailored for linear performance functions that represent weighted shortfalls from cumulative demand levels. A lexicographic minimax algorithm that can employ any of these algorithms is also presented.

Next, this chapter examines the allocation of nonstorable resources. These resources include production facilities and workforce where unused capacity in any given period is lost, and perishable resources like certain medications and fresh food that need to be discarded at the end of a period. The multiperiod model with nonstorable resources is formulated with a nonseparable objective function, similar to Problem L-RESOURCE with a nonseparable objective function described in Section 3.4.

Finally, we examine multiperiod resource allocation models with substitutable resources. As expected, such models and the corresponding algorithms combine elements of the models and algorithms described in Chapter 4 and in this chapter.

1.5.5 Chapter 6: Equitable Network Resource Allocation

Chapter 6 presents allocation of network resources in communication and computer networks, though some of the models also apply to other areas, for example, transportation.

The chapter first considers network flow problems with demands between multiple node pairs, where the throughput for a given node pair is the total flow routed between these nodes. Such problems are referred to in the literature as multicommodity network flow problems.

The simplest problem considers a network where the demand between each node pair is routed on a single fixed path. The problem with a lexicographic maximin objective is then essentially the same as Problem L-RESOURCE of Chapter 3. A simple algorithm for performance functions that represent weighted throughputs is presented. A more challenging problem arises when the demand between each node pair may be routed on multiple paths and the throughput is then the sum of flows along these paths (recall the example in Section 1.3.2). When the set of feasible paths for each node pair is limited and given as input, this problem is the same as that discussed in Section 3.4 for a nonseparable objective function. Thus, for performance functions that represent weighted throughputs, the lexicographic maximin solution is computed by repeatedly solving linear programming problems, where after each such problem is solved, some demands are identified as saturated. However, the actual flows of saturated demands across the multiple paths are only determined at the last iteration, once all demands are saturated. When the set of paths for each demand is not specified as input, the problem is even more challenging since the number of feasible paths may be prohibitively large. In that case, some column generation method (a well-studied topic in large-scale

optimization) needs to be incorporated within the linear programming algorithm. When the performance functions are concave, the algorithm must repeatedly solve convex optimization problems instead of linear programming problems.

Next, we consider equitable bandwidth allocation for content distribution through a network composed of one or more tree topologies, where a server at the root of a tree stores multiple programs that are broadcasted throughout the tree. Primary application areas include VOD for home entertainment, remote learning and training, video conferencing, and news on demand. Each link of the network carries at most one copy of any program; thus, the problem is quite different from the standard multicommodity network flow problems. Furthermore, the bandwidth allocated to a program may decrease when moving away from the server, but may not increase. This results in treelike ordering constraints in the formulation as the bandwidth allocated to a program on a link must be at least as large as the bandwidth allocated to that program on all successor links. Section 1.3.4 presents an example of a single tree network for content distribution.

We examine two problems. The first problem assumes that the performance function for a specific program is the same at all nodes. The resulting lexicographic maximin algorithm is then a relatively simple extension of that for Problem L-RESOURCE of Chapter 3, where the extensions are needed to enforce the treelike ordering constraints. The second problem allows node-dependent performance functions for each program. The previous algorithm cannot be extended to handle this case. Instead, a maximin algorithm that employs a numerical search is presented. This algorithm extends one of the algorithms of Chapter 5 for multiperiod problems with storable resources. The maximin algorithm is then incorporated into a lexicographic maximin algorithm.

1.5.6 Chapter 7: Equitable Resource Allocation with Integer Decisions

This chapter covers equitable resource allocation models with integer decision variables. Thus, unlike in all problems discussed in Chapters 3–6, the feasible region is not convex. This nonconvexity forces the development of different solution approaches since the solution of a minimax problem does not provide obvious guidance as to which activity levels can be fixed at their lexicographic minimax value.

The chapter starts by examining the challenges through Problem L-RESOURCE with integer decision variables. An efficient algorithm for solving the minimax problem is presented; however, as will be shown, it cannot readily be extended to solve the lexicographic minimax problem. A simple algorithm, based on marginal allocations, is described for a special case of the lexico-

graphic minimax problem with a single resource constraint. This problem is related to the problem of fair apportionment of seats in a legislative body.

Next, the emergency facility location problem, discussed in Section 1.3.5, is examined. The problem can be converted to a lexicographic minimization problem, where the rth term in the objective function is the number of occurrences of the rth worst possible distinct outcome. The optimal solution is then obtained by minimizing the first term in the objective function, followed by minimizing the second term without increasing the first term, and so forth. This lexicographic minimization requires repeatedly solving mixed integer programming problems. Although constructing the counting functions for $r = 1, 2, \ldots$ is easy for the facility location problem with a negligible increase in the problem size, in general, it can only be done at the expense of adding many auxiliary continuous variables and constraints. We present details of the general approach. However, this solution approach is practical only if the number of possible distinct outcomes is relatively small.

Next, solution approaches for problems with a large, possibly infinite, number of distinct outcomes are presented. Although such problems can be found in many application areas, much of the work has been done in the context of communication networks, where consideration of integer decision variable is often important. The key idea consists of expressing the sum of the k worst performance function values for a given solution as an optimization problem. As a result, the original lexicographic maximin (or minimax) problem can be converted to a standard lexicographic optimization problem, again, at the expense of adding many auxiliary continuous variables and constraints. Still, although this approach can be used for many problems, the computational effort may become prohibitively large since it may require the repeated solution of large mixed integer programming problems. To that end, several approximation approaches are also presented.

1.6 CONCLUDING REMARKS AND LITERATURE REVIEW

Resource allocation models are concerned with the allocation of limited resources among numerous activities. This book focuses on resource allocation models that can be adapted to diverse application areas, and for which efficient, elegant solution methodologies can be developed because of their special mathematical structure. In particular, most of the material covers resource allocation models with a lexicographic minimax (or a lexicographic maximin) objective function. We often refer to these models as *equitable resource allocation models* with the understanding that these models use lexicographic minimax (or maximin) objective functions.

It is important to note that linear programming, the most widely used operations research methodology, addresses numerous resource allocation problems. Many books have been written on this topic. We refer to just three:

the seminal linear programming book by Dantzig [Dan63]; the book by Lasdon [Las70], which presents algorithms for large-scale problems; and the more recent book by Vanderbei [Van08] (first edition published in 1997). Much of the work on resource allocation models with special mathematical structures has been inspired by the initial work of Koopman [Koo53, Koo56a, Koo56b, Koo57] on search effort distribution. Over the decades, operations research has demonstrated many success stories as documented, for example, in the Committee on the Next Decade in Operations Research (CONDOR) [CO88] and in Luss and Rosenwein [LR97]. Much of the work that led to these successes includes elements of resource allocation models and solution methodologies.

As work on resource allocation models with special mathematical structure grew along with better solution methodologies, there was an obvious need for books and expository papers that would be a source for education and guidance. Mjelde [Mje83a] published the first book that covers a large number of such models. Ibaraki and Katoh [IK88] wrote an excellent book that covers much of the work until that date primarily for models with a single resource constraint for both continuous and integer decision variables. Katoh and Ibaraki [KI98] continued that work in a detailed survey with an emphasis on models with integer variables. Luss and Smith [LS86] published the first paper on lexicographic minimax approach for resource allocation problems with continuous variables and multiple resource constraints (Problem L-RESOURCE). Luss [Lus99] presented an expository paper on equitable resource allocations using a lexicographic minimax (or lexicographic maximin) approach. Pioro and Medhi [PM04, chapters 8 and 13] described models and algorithms for equitable resource allocations in networks.

Significant work has been devoted to the understanding of different notions of fairness, starting with the work of Pigou [Pig12] in 1912 and Dalton [Dal20] in 1920 in the field of economics. Rawls [Raw71] presented a comprehensive theory of justice, which among other implications protects the most unfortunate individuals from "the greater value to society" (as in a minimax objective). The literature on multiobjective optimization models is quite rich; see, for example, Isermann [Ise82] and surveys by White [Whi90], Ehrgott and Gandibleux [EG00], and Marler and Arora [MA04]. Some multiobjective optimization models assume that the objectives are incomparable, and thus, the performance functions have no obvious basis for comparison. One solution approach to such models uses lexicographic optimization, where an objective function vector consists of multiple criteria, which are optimized sequentially in a predetermined order.

This book focuses on resource allocation applications where the different objectives are comparable, and it is important to measure and achieve some equitable allocation of the resources. To that end, the concept of a lexicographic minimax (or lexicographic maximin) objective, a natural extension of the widely used minimax (or maximin) objective, is introduced. A similar notion is known in cooperative games theory as the nucleolus allocation;

see, for example, Schmeidler [Sch69], Potters and Tijs [PT92], and Kern and Paulusma [KP03]. Many references, including Ogryczak [Ogr97] and Kostreva and Ogryczak [KO99], describe preference relations for equitable solutions (shown in Section 1.4) that are satisfied by lexicographic minimax optimization. As mentioned earlier, lexicographic minimax solutions are also pareto-optimal and, thus, are called *equitably efficient*. Indeed, lexicographic minimax solutions are sometimes referred to as the "most equitable solutions" (see, e.g., Kostreva and Ogryczak [KO99]). Bertsekas and Gallager [BG92, chapter 6] introduced the max-min fair flow solution for networks with flows over a single fixed route (it appeared in the first edition published in 1987). Radunovic and Le Boudec [RLB07] and Nace and Pioro [NP08] discussed the max-min fairness criterion and its relation to lexicographic maximin optimization.

This book presents many problems for which effective lexicographic minimax algorithms do exist. For example, for certain classes of performance functions, the algorithms can solve large-scale problems in the format of Problem L-RESOURCE very fast. For more complicated problems, for example, with substitutable resources and multiperiod allocations, more involved algorithms, which leverage of the algorithms for Problem L-RESOURCE, are available; see Luss [Lus99] and references therein. Algorithms for problems with a non-separable objective function are more time-consuming and, typically, require solving multiple linear or nonlinear optimization problems. Such problems are common in communication network applications; see, for example, Pioro and Medhi [PM04, chapters 8 and 13] and Nace and Pioro [NP08]. Next, consider problems with integer decision variables. Such problems are significantly more difficult, and, in general, are converted to lexicographic minimization (or maximization) problems at the expense of adding many auxiliary variables and constraints. This lexicographic optimization problem is solved by repeatedly solving large mixed integer programming problems. In one approach, the lexicographic objective is composed of counting functions that count the number of occurrences of distinct outcomes; see, for example, Ogryczak, Pioro, and Tomaszewski [OPT05]. In a different approach, the lexicographic objective is composed of multiple criteria, where the kth criterion is the sum of the k worst performance function values; see, for example, Ogryczak, Wierzbicki, and Milewski [OWM08]. Although these approaches are very useful, large problems, such as those encountered, for example, in large communication networks with integer decision variables, may require prohibitively large computing effort. Hence, various algorithms that provide approximations to the lexicographic minimax solutions have been proposed.

Wierzbicki [Wie82] and Wierzbicki, Makowski, and Wessels [WMW00] presented interactive methods that provide a range of equitably efficient solutions based on reservation and aspiration levels for each of the activities. The reservation levels are the required activity levels, whereas the aspiration levels are the desired levels, commonly referred to as reference points. Depending on the aspiration and reservation levels, a utility function is constructed. If the

solution achieved for the utility function is not acceptable to a decision maker, the reservation and aspiration levels are modified and a new utility function is constructed and a new problem is solved; otherwise, the method terminates. Many successful applications are reported in Wierzbicki, Makowski, and Wessels [WMW00] and Kaleta et al. [KOTZ03], as one may provide the decision maker with numerous appealing alternatives. Gardiner and Steuer [GS94a, GS94b] examined a variety of interactive methods for problems with multiple objectives and provide a unified approach to these methods.

Kelly, Maulloo, and Tan [KMT98] employed a logarithmic utility function that leads to proportional fairness. Mo and Walrand [MW00] presented the α-fair utility function, which includes a wide spectrum of fairness criteria, depending on the selected value for α. Lan et al. [LKCS10] presented an even more general axiomatic theory of fairness in network resource allocation, which includes, among others, the α-fair utility notion as a special case. The interested reader may consult the references listed in these papers.

It is important to understand and quantify the price of fairness relative to a fully efficient allocation that maximizes the sum of all performance functions. Tang, Wang, and Low [TWL04] examined the price of fairness for network flow problems using the α-fair utility function. Bertsimas, Farias, and Trichakis [BFT11] examined the price of fairness for a broad family of problems, focusing on proportional fairness and max-min fairness. It should be noted that in various applications, solutions that are perceived as unfair to some are doomed to fail in practice. For example, the majority of proposals for efficient allocation schemes of takeoff and landing "slots" at airports (an expensive scarce resource), which benefit the overall system at the expense of some airlines, are often discarded as impractical.

We conclude with a sample of references used for the applications described in Section 1.3. Many more references will be mentioned throughout the book.

1.6.1 Equitable Allocation of High-Tech Components

Luss and Smith [LS86] presented the first paper on this application with a lexicographic minimax objective. Related papers include Tang [Tan88] and King [Kin89]. The latter presents an interactive solution approach for production planning that repeatedly solves problems in the format of Problem L-RESOURCE.

1.6.2 Equitable Throughput in Communication and Computer Networks

Bertsekas and Gallager [BG92, chapter 6] presented a simple max-min fair solution for networks with a single fixed route per demand. Pioro and Medhi [PM04, chapters 8 and 13] presented a variety of equitable flow models for

networks, including models where the flow between a node pair may be routed on multiple paths. Ogryczak, Pioro, and Tomaszewski [OPT05] and Ogryczak, Wierzbicki, and Milewski [OWM08] presented various models with continuous and integer decision variables.

1.6.3 Point-to-Point Throughput Estimation in Networks

Kruithof [Kru37] presented the first paper on this topic; see Krupp [Kru79]. Since then, numerous solution approaches have been proposed for different variants of this problem. Luss and Vakhutinsky [LV01] presented a resource allocation model with a lexicographic minimax objective that computes estimated point-to-point throughputs based on load measurements, and Luss [Lus05] described a similar model for cellular wireless networks.

1.6.4 Equitable Bandwidth Allocation for Content Distribution

Sarkar and Tassiulas [ST00] presented the first paper on this application for an equitable throughput objective. The formulation in Section 1.3.4 is quite different and follows Luss [Lus08, Lus10]. Lee, Moon, and Cho [LMC04] and Sarkar and Tassiulas [ST02] addressed this problem with integer bandwidth allocation decisions.

1.6.5 Equitable Location of Emergency Facilities

Ogryczak [Ogr97] presented an algorithm that computes a lexicographic minimax solution by minimizing first the number of occurrences of the worst possible outcome, followed by minimizing the number of occurrences of the second worst possible outcome, and so forth. Ogryczak [Ogr99, Ogr00] also presented other approaches with fairness criteria to the location problem.

1.6.6 Other Applications

The broad applicability of equitable resource allocation models is further demonstrated through a sample of references in diverse application areas. These include the following:

- *Air Traffic Management*: Sherali, Staats, and Trani [SST03, SST06], Vossen and Ball [VB06], and Bertsimas, Lulli, and Odoni [BLO11].
- *Congestion Control in Sensor Network and Sensor Location Decisions*: Chen, Fang, and Xia [CFX07] and Neidhardt, Luss, and Krishnan [NLK08].
- *Municipal Solid Waste Management*: Erkut et al. [EKPT08].
- *Visual Quality of Video Coded Pictures*: Hoang, Linzer, and Vitter [HLV97].

- *Health Services, for Example, Evaluation of Radiation Dose Distribution*: Svahn, Peterson, and Hansson [SPH06].
- *Search Effort*: Koopman [Koo53, Koo56a, Koo56b, Koo57].
- *Water Rights Allocation*: Wang, Fang, and Hipel [WFH07, WFH08].
- *Military Applications*: Danskin [Dan67], Jaiswal [Jai97], Newman et al. [NRSB11], and Golany et al. [GKPR12].
- *Final Assembly Sequencing in Production Systems*: Monden [Mon98] (first edition published in 1983) and Groeflin et al. [GLRW89].