# Linear Programming

Robert W. Llewellyn

# LINEAR PROGRAMMING

**Robert W. Llewellyn**
North Carolina State College

HOLT, RINEHART AND WINSTON
New York • Chicago • San Francisco • Toronto • London

90123 38 98765


Library of Congress Catalog Card Number: 64–10124
**03-041690-6**

Printed in the United States of America

# LINEAR PROGRAMMING

# preface

THE OBJECTIVE OF THIS BOOK is to present a reasonably complete account of linear programming. The mathematical development is based on the theory of simultaneous linear equations, without the usual notions from the mathematics of vector spaces. However, the level of mathematical reasoning is such that a background of two terms of college-level mathematics is desirable.

The mathematical developments have been carried out in considerable detail. In almost all instances, each new method is immediately illustrated by a numerical calculation that shows how the previous theory applies.

The transportation problem and its solution method are discussed in Chapter 2, with the simplex method delayed until Chapter 4. The transportation method can be developed independently of the simplex method; because the former is easier to understand and easier to apply, the latter then becomes easier to grasp. The arguments used to develop the simplex method parallel those used to develop the transportation method. By going over the same arguments in two different contexts, the developments reinforce each other.

The treatment of topics in Chapters 1–8 is conventional, although the details are quite different from those used in other texts. In Chapter 8, emphasis is placed on the dual theorems. (It is recommended that all of the first eight chapters be included in a college course.)

In Chapter 9, primal-dual methods are discussed. These topics are rather specialized, but of great importance to anyone working intensively in the field who needs the most efficient solution methods. The material of Chapter 10 treats sensitivity analysis, the expansion problem, and parametric programming.

Chapter 11 deals with problems that have upper-bound and/or integral constraints on their variables. Chapter 12 involves applications and methods of putting those problems into the linear-programming format that do not appear at first glance to be linear-programming problems.

The topic of Chapter 13 is the correspondence between game theory and linear programming. Chapter 14 covers upper-bound methods in the transportation problem, the assignment problem, and some simple network applications.

The author is grateful to his colleagues at North Carolina State College for their encouragement during the preparation of the manuscript, and especially to Mr. H. A. Knappenberger, who used much of the material in early form and made many valuable suggestions. The author also wishes to express his appreciation to Mrs. Adele Covington, who typed the manuscript.

R. W. LLEWELLYN

Raleigh, North Carolina
*December 1963*

# contents

# chapter 1

# *INTRODUCTION*

## 1-1 BACKGROUND

LINEAR PROGRAMMING is a product of modern mathematics and is, at this writing, about fifteen years old. The problem area has been of interest for a longer period; there is a rather large literature in both mathematics and economics in this field dating back into the 1920's. Dr. George B. Dantzig published his first paper on the simplex method in 1947. Since that time progress in the field has been rapid. The first applications were military in nature, but it was not long before it became apparent that there were important industrial applications as well.

Linear programming problems have the following framework.

1) There is some *objective* to be attained, such as maximum profit, minimum cost, or minimum elapsed time, of the system being studied.

2) There are a large number of *variables* to be handled simultaneously. The variables may be products, machine-hours, man-hours, money, floor space, or other factors, depending on the problem. There are usually several kinds of variables in a problem. Some of these are *outputs* of the system (such as products), while others are *inputs* to the system (such as man-hours). The latter are sometimes called the *resources.*

3) There are many *interactions* between the variables. A typical problem is that of determining the best product mix for a production period. Here we are trying to determine which products to manufacture, out of a list of potential products, together with the optimal quantity of each, so as to maximize the total profit received from all products over some stated production period. The interactions arise from the fact that if we have limited resources and manufacture a stated amount of product $A$, there are then fewer resources available for the production of products $B$, $C$, $D$, etc. The products, in a sense, compete for the available resources. The linear programming model can be used to determine how to resolve this conflict so as to obtain the most profitable production program. Obviously, the unit profit obtainable from each potential product is of significance in determining how this competition should be resolved.

4) Most linear programming problems are also characterized by the presence of objectives that *conflict* with the principal objective of the problem. In the product-mix case, for example, the manufacturer may specify that *at least* a certain amount of one of the products be made, regardless of the effect on profit. The objective competing here with that of maximizing profits may be to fulfill an order already received and accepted.

Thus linear programming tends to be associated with *complex situations*, many *interacting variables*, and *competing objectives* along with the optimization of some criteria of the effectiveness of the system. The interactions of variables and competition of objectives are characteristic of many industrial situations. Indeed, they are characteristic of all economic systems, which fact explains the early theoretical interest in this type of problem among economists. It is natural, then, for industry to find that many of its most important problems can be solved by linear programming methods. It is incorrect to assume that all industrial problems involving these elements of interaction and conflict can be handled with linear programming methods, however. The word *linear* means just what it says; problems can be put into the linear programming model only if the algebraic relationships between the variables are linear or can be closely approximated by first-order equations. If this condition is violated, other techniques beyond the scope of this book must be used.

A few examples of the use of linear programming, other than the product-mix problem already mentioned, are

1. *The diet problem.* The *diet* problem gets its name from the fact that an early application was to determine the most economical human diet. In its most common industrial form, it consists of determining the most economical mixture of raw materials that will result in a product with a desired chemical formula. The importance of the problem lies in the fact that market prices and stocks of the various materials change from time to time. The most economical mix one week may not be the most economical, or even possible, the next.

2. *The machine-loading problem.* In *machine-loading*, we are concerned with assigning production jobs to machines in such a manner as to minimize production costs over the entire schedule of a department or machine center for a production period. The typical plant situation, at least in job-lot or intermittent plants, is that some machines are most efficient for many of the scheduled orders, and that the production for the period cannot be accomplished by simply assigning each order to the most efficient machine. Some jobs must be assigned to a second best or a third

best alternative. Linear programming resolves this conflict, providing the over-all machine time available is adequate, so as to use the machines most efficiently when *all* orders are considered simultaneously.

3. *The production-scheduling problem.* There are several problems that come under the heading of *production scheduling* and we will discuss only one here. Suppose a company makes a single product on a production line but that the rate at which the line is run can be varied. We shall also assume that the expected sales pattern over the production season is known. We will not ordinarily produce the exact amount in each production period to satisfy the demands of that period because this would involve wide variations in production over the season and is not economical. Instead, we will attempt to smooth production, in relation to sales, and allow an inventory to absorb the difference between production and sales quantities. It costs money to store the product and it also costs money to change the rate of production. Linear programming can be used to determine the production schedule over the season that will minimize the sum of the inventory carrying costs and the changeover costs.

4. *The transportation problem.* In the *transportation* problem we are concerned with a product that is stored in a number of *origins*, perhaps the plants in which it was made, and needed in a number of *destinations*, perhaps jobbers or distribution warehouses. It is assumed that we know the quantity of the product available in each origin and the quantity needed at each destination. We are also given the unit cost of shipping the product from each origin to each destination. The objective of the problem is to determine the quantity to ship from each origin to each destination so as to minimize the sum of the shipping costs. Because of special properties of the problem, a special method can be used to solve it. We will study this problem in Chapter 2 and again in Chapter 14.

The examples cited here are only a few of the problems to which linear programming has been applied in industry. It is likely that more and more applications will be discovered as larger numbers of qualified persons are employed for work in the fields in which the problems are encountered. One of the most important facts about the problems most suited to this type of treatment is that they tend to be repetitive; that is, the problems are of an operating nature and thus must be solved periodically, say daily or weekly. This is fortunate, for the typical problem is so large that the cost of the analysis necessary to develop a model for it is greater than the savings resulting from a single application. But if the same problem, except for minor variations in detail, is to be solved on a periodic basis, the initial cost of analysis becomes small per solution obtained. It should also be noted here that almost all practical applications are so large that the use of digital computers is mandatory. In fact, it is likely that if the emergence of linear programming were not paralleled by the appearance of digital computers, progress in linear programming would have been much less rapid and there would be little need for books such as this, it being aimed at the practitioner rather than the mathematician.

In order that the reader appreciate the importance of the earlier comments on the interactions and competing objectives in linear programming, we will present an example of a type of analysis that does not involve these factors. Suppose a product is expected to have a yearly demand of $R$ units, that it costs $P$ dollars to prepare for a production run, $C$ dollars per unit to make the product, and $S$ dollars to store one unit of the product for one year, the storage costs including the return expected from capital in the form of interest. If $Q$ is the amount we will produce in one production run and $Y$ is the total cost of a year's production making $Q$ units each time (it is assumed that the rate of sales or use of the product is constant over the year), then

$$\text{setup costs per year} = P\frac{R}{Q} \tag{1.1}$$

$$\text{production costs per year} = CR \tag{1.2}$$

$$\text{storage costs per year} = \frac{SQ}{2} \tag{1.3}$$

$$Y = P\frac{R}{Q} + CR + \frac{SQ}{2} \tag{1.4}$$

We assume that $Q$ is a continuous variable. For both small $Q$ and large $Q$, $Y$ is large. To find the $Q$ for which $Y$ is a minimum, we determine where $dY/dQ$ vanishes, or

$$\frac{dY}{dQ} = -\frac{PR}{Q_0^2} + \frac{S}{2} = 0 \tag{1.5}$$

$$Q_0 = \sqrt{\frac{2PR}{S}} \tag{1.6}$$

where $Q_0$ is called the "economic lot size." It is possible to express the quantities, particularly $S$, in more detailed form and get more precise versions of (1.6). The point we wish to make here is that:

1) Many authorities from academic and engineering fields feel that equations like (1.6) are of great value in minimizing the cost of production programs.

2) Industrial personnel frequently find equations like (1.6) of little use and refuse to use the economic lot-size concept.

The answer to the dilemma is that the mathematical model used to derive (1.6) may be incomplete. In many plants the storage space is not sufficient to accommodate the goods if production is always in economic lot sizes. In others, sufficient capital is not available to finance the inventory if that much of each product is manufactured each time it is made. In still others, where capital for storage is available, management may prefer to divert it to other objectives, such as financing new plants. When applied to some plants, (1.6) fails to recognize *limitations on resources;* in other applications it fails to recognize and include the effect of *other competing objectives.* Out of these considerations has developed a whole new area of research in inventory control in which these limitations of resources and the effect of other competing objectives are included. Some of this research utilizes linear-programming methods.

## 1-2 LINEAR PROGRAMMING

The basic problem in linear programming is merely to maximize (or minimize) a linear function of the form

$$z = c_1x_1 + c_2x_2 + \cdots + c_nx_n \tag{1.7}$$

Now, if we assume, as we did in equation (1.4), that the variables can assume any values, the problem is trivial, as we can let each variable that has a positive coefficient become as large as we please and the value of the function then becomes arbitrarily large. If we are minimizing, we can let each variable that has a negative coefficient become as large as we please and the function assumes an arbitrarily large negative value. Or, still minimizing, if all the coefficients are positive, we can let each variable become zero and then the value of the function becomes zero. In any event, the problem is trivial unless there are some restrictions on the variables. We will discuss these restrictions presently.

First, it should be noted that we did not mention letting any of the variables become negative. In linear programming the variables are usually restrained from taking on negative values. Suppose the function is

$$z = 3x_1 + 2x_2 + 4x_3 \tag{1.8}$$

where $x_1$, $x_2$, and $x_3$ represent products to be made in the factory. If $x_1$ represents units produced, then $x_1 = -10$ must represent units brought back from customers, disassembled, and converted back to raw materials. This is not only a poor business policy, it is technologically impossible in a great many cases. Linear-programming problems are from a family of processes an economist would call *irreversible.* As practical problems are presented later, the reader will see that in nearly every case the context of the problem does not permit the assignment of negative values to the variables. This is made a formal requirement of the linear-programming problem as follows. We will use the index $j$ to number the variables and $n$ to indicate their total number. Since the variables are permitted to assume the value of zero, the restriction is called the *nonnegativity restriction* on the $x_j$, or

$$x_j \geq 0, \quad j = 1, 2, \cdots, n \tag{1.9}$$

It will be assumed throughout this book that (1.9) applies unless a special note is made to relax the requirement.

The other restrictions that may be on the variables are illustrated by the following examples.

1) $x_1 \leq 3$. This restriction, plus (1.9), permits $x_1$ to assume any value between zero and 3, including both extremes; $x_1$ is still a variable, but only within stated limits.

2) $x_1 \geq 3$. This restriction permits $x_1$ to assume any value from 3 on up. The restriction of (1.9) applied to $x_1$ is now redundant, since the requirement that $x_1$ be at least 3 is stronger than the requirement that it be at least zero.

3) $x_1 + 2x_2 + 8x_3 = 4$. This is a combined restriction on three variables at once and permits $x_1$, $x_2$, and $x_3$ to assume any values such that, when multiplied by their coefficients, the sum equals 4. We will always assume, however, that the variables are also restrained from being non-negative by (1.9).

4) $x_1 + x_2 - x_3 \leq 5$. This is also a combined restraint on several variables and the variables $x_1$, $x_2$, and $x_3$, while being nonnegative, must sum, algebraically, to less than or equal to 5. This permits more freedom for the variables than the equality type of restriction. This restriction also illustrates that while the *variables* are constrained to be nonnegative, their *coefficients* may be negative in these constraints.

5) $2x_1 - x_2 + 3x_3 \geq 9$. This is interpreted as is the preceding example, except that now any combination of the variables, when multiplied by their coefficients, must sum to 9 or more.

6) $x_1 = x_2$. This restrains $x_1$ from taking on any value other than that taken on by $x_2$ and vice versa. This will usually be rewritten as $x_1 - x_2 = 0$.

## 1-3 SIMULTANEOUS EQUATIONS

We will see that linear programming is concerned with solutions to simultaneous linear equations. These equations arise from the restrictions on the variables. Yet the restrictions are frequently stated as inequalities rather than as equations. The inequalities can be converted to equations as follows.

*Type* I. We will call the type of inequality where the sign is read "less than or equal to," the "Type-I" inequality. This is a convenience, since frequent repetition of the phrase "less than or equal to" is awkward. This designation is not common in mathematics, but will apply throughout this book.

A Type-I restriction is of the form

$$k_1x_1 + k_2x_2 + k_3x_3 \leq b \tag{1.10}$$

where $k_1$, $k_2$, and $k_3$ are constants. A Type-I inequality is converted to an equation by *adding* a nonnegative slack variable. Thus, we convert (1.10) to

$$k_1x_1 + k_2x_2 + k_3x_3 + x_4 = b \tag{1.11}$$

Here $x_4$ is called a *slack variable* and assumes whatever value is necessary for the equation to be satisfied. For example, if $x_1 = x_2 = x_3 = 0$ in (1.11), then $x_4$ must equal $b$. The slack variable will be considered as under the nonnegativity restriction (1.9) and is one of the $x_j$ variables, $j = 1, 2, \cdots, n$.

*Type* II. We will likewise refer to the "more than or equal to" inequality as the "Type-II" restriction. It is of the form

$$k_1x_1 + k_2x_2 + k_3x_3 \geq b \tag{1.12}$$

To convert a Type-II inequality to an equation we must *subtract* a nonnegative variable. Thus, (1.12) becomes

$$k_1x_1 + k_2x_2 + k_3x_3 - x_4 = b \tag{1.13}$$

where $x_4$, again included in the nonnegativity restriction, is a slack variable that permits the equation to be satisfied.

For example,

$$3x_1 + x_2 - 2x_3 \geq 10 \tag{1.14}$$

converts to

$$3x_1 + x_2 - 2x_3 - x_4 = 10 \tag{1.15}$$

If $x_1 = 5$, $x_2 = 10$, and $x_3 = 5$, then $x_4$ must equal 5.

The reader should note that the treatment of inequalities is different in linear programming than it is in many other mathematical applications. Ordinarily when the nonnegativity restriction is not imposed, a slack variable is added to either type of restriction to convert it to an equation.