Introduction to

DYNAMIC PROGRAMMING

George L. Nemhauser
Department of Operations Research and Industrial Engineering
The Johns Hopkins University

JOHN WILEY AND SONS, INC.
New York • London • Sydney

FIRST CORRECTED PRINTING, OCTOBER, 1967

Library of Congress Catalog Card Number: 66-21046
Printed in the United States of America

Introduction to Dynamic Programming

SERIES IN DECISION AND CONTROL

Ronald A. Howard

INTRODUCTION TO DYNAMIC PROGRAMMING

by George L. Nemhauser

DYNAMIC PROBABILISTIC SYSTEMS (In Preparation)

by Ronald A. Howard

TO

Loring G. Mitten

Preface

Dynamic programming is used to solve complex optimization problems. This book presents the theory and computational aspects of dynamic programming. It is an applied book, designed for operations researchers, management scientists, statisticians, engineers, and social scientists.

Scientific decision-making involves model building and then solving the model to determine an optimal solution. Many models, encompassing different disciplines and areas of application, are amenable to solution by dynamic programming. These models contain many decision variables and have a mathematical structure which is such that calculations of the optimal decisions can be done sequentially. When and how the calculations can be done sequentially is the essence of dynamic programming. Sequential optimization loosely means determining the optimal decisions one at a time. Often the ability to determine decisions one at a time makes a problem computationally feasible.

Extensive applications have been made in inventory theory, allocation problems, control theory, search theory, and chemical engineering design. By identifying the mathematical structures amenable to dynamic programming analysis, it is hoped that new applications will be developed. This problem is the subject of Chapter II. It is preceded by a brief discussion of model building, the dynamic programming approach, and optimization in Chapter I. Chapter II presents the basic approach of multistage problem solving and when it can be used in optimization. By knowing when it is theoretically possible to use dynamic programming, an analyst can decide whether it is possible to solve his problem by dynamic programming. But, knowing that something is possible is different from knowing exactly how to do it.

Chapters III and IV are the "how-to-do-it" chapters. Basic computations are the subject of Chapter III—how the dynamic programming formulation is obtained, how the computations are organized, the preparation of flow charts for computers, the data requirements, and sensitivity analysis. The exercises are especially essential in Chapter III. They demonstrate the basic ideas of dynamic programming formulation and solution. It is absolutely necessary to solve problems to understand dynamic programming. Methods for doing computations as efficiently as possible are given in Chapter IV. This is most crucial when expensive computer time is used.

Chapter V extends the results obtained for deterministic multistage decision models to stochastic and competitive models. In Chapter VI the usual assumption about the serial structure of adjoining stages is removed to extend the analysis to processes with branches and feedback loops. Models with an infinite number of decisions are discussed in Chapter VII. The relationship between dynamic programming and the calculus of variations is revealed. Some general conclusions and a discussion of applications are given in Chapter VIII.

Almost every new idea introduced is illustrated with a detailed analysis of one or more examples. The form of the examples assumes that a model has already been constructed, so that attention can be given to its dynamic programming formulation and solution. In this framework, one can relate his own problems to the problems in the text by adding the necessary context to the examples. Furthermore, organizing the problems according to their mathematical structure will be of great advantage to those interested in developing new applications.

The development of dynamic programming is almost exclusively due to Richard Bellman and his colleagues at the Rand Corporation. His books and papers furnished a large fraction of the source material for this text. Professor L. G. Mitten of Northwestern University introduced me to dynamic programming. His interest stimulated mine. I am thankful to him for encouragement and for valuable suggestions on the organization and technical content of the book. I hope the book makes him proud. Dr. William W. Hardgrave read a draft of the manuscript in detail. His comments were invaluable in transforming the manuscript from draft to final copy. My wife, Ellen, has helped immensely with style and grammar—a difficult task considering her lack of interest in the subject matter. All of the several reviews obtained by John Wiley have helped to improve the book. A draft of the book was used in a one-semester two-hour-per-week course at Johns Hopkins which I taught jointly with Dr. Mandell Bellmore. I am grateful to him and the students who found numerous errors while suffering through the rough draft. I would like to express my appreciation to Mrs. Helen Macaulay of Johns Hopkins for her excellent typing of part of the first draft and the entire second draft of the manuscript, and to Miss Jane Shaw of Leeds University, U. K., for typing a considerable portion of the first draft. I am also grateful to Michael Magazine for reading the galley proofs and assisting with the preparation of the index.

George L. Nemhauser

Baltimore, Maryland
April 1966

Contents

I

Introduction

1. Background

Dynamic Programming is an approach to optimization. Optimization means finding a best solution among several feasible alternatives. The term "a best solution" is used because there may be more than one optimal solution.

The representation of a problem in abstract or symbolic form is known as a *mathematical model*. Characterizing optimization problems by mathematical models goes back to the Greeks, if not further. Attributed to Greek mathematics is the solution of problems such as finding the geometric figure of minimum perimeter that encloses a given area.

Theories of optimization existed long before the development of the calculus. Nevertheless, the formal development of optimization theory came from the calculus. After the invention of the calculus, mathematicians worked actively on optimization problems. The theory was developed for mathematical models containing continuous variables and differentiable functions. Many of the problems studied were of geometrical background. Although the theory provided solution procedures for problems with several variables, the theory was not adequate to deal computationally with models containing a very large number of variables. However, a few variables were generally sufficient to characterize most of the geometrical problems of interest then. The classical development of optimization theory through the calculus was essentially complete by the end of the nineteenth century. A good exposition is contained in *Theory of Maxima and Minima* by Hancock [35].

In the 1940's there was a reawakening and change of direction in the study of optimization theory. This renaissance was stimulated by the war effort. Two parallel but interrelated occurrences are especially significant;

the work of scientists and mathematicians on military operational problems, and the invention and development of the digital computer.

The scientific approach to military problems, and then, after the war, to industrial and other institutional problems, became the field of study known as *operations research*.† The formulation and solution of mathematical models of optimization is an integral part of operations research. These models of complex logistic, production, and distribution systems are generally characterized by a large number of variables, and are often of a form not amenable to solution by the calculus.

Pioneer operations analysts simultaneously developed models and solution techniques. Their notable successes were partly due to the rapid evolution of high-speed digital computers. Given a machine that could do thousands of calculations per second, it became practical to think about solving problems containing hundreds or even thousands of variables. This realization stimulated the study of iterative optimization schemes and eventually led to the development of linear and nonlinear programming, dynamic programming, and various search methods.

The evolution of these methods and their refinement may be thought of as the renaissance of optimization theory. The motivation for this regenerated interest stems from operations research. Thus modern optimization theory is usually studied in conjunction with operations research. However, it can and has been applied to problems completely within the domain of traditional disciplines. It is possible to study the methodology without making reference to digital computers. However, as the success of the methods depends largely on digital computers, we shall consider the role of the computer.

2. Mathematical Models of Decision Making‡

A mathematical model is a symbolic representation of relations among the factors in a problem of decision making. The basic components of the model are:

1. The variables $D = (d_1, d_2, \ldots, d_n)$—those factors that can be manipulated to achieve the desired objective. These variables are commonly referred to as independent or decision variables.

2. The parameters $Y = (y_1, y_2, \ldots, y_p)$—those factors that affect the objective but are not controllable.

3. The measure of effectiveness (R)—the value, utility, or return as-

† Some general references on operations research are Churchman et al. [23], Flagle et al. [30], and Saaty [60].

‡ Ackoff [1] contains an excellent discussion of mathematical decision models.

sociated with particular values of the decision variables and parameters. The measure of effectiveness, alternatively called the utility measure, criterion function, objective function, or return function, is a real-valued function of the decision variables and parameters, which can be represented as

$$R = R(D, Y)$$

There is a wide variety of commonly used measures of utility, such as cost, profit, rate of return. It will be assumed that a specific measure of effectiveness can always be chosen that will adequately reflect the important differences among different values of the decision variables.

4. *The region of feasibility* (*S*)—in most circumstances the decision variables are limited in the values they can assume. These limitations are generally given by specifying a region of feasibility or constraint set (*S*). The feasible values for the decision variables must be contained in the set *S*, that is, ($D \in S$). Sometimes it is possible to represent all or a part of the constraint set by equations and/or inequalities of the form

$$g_i(D) \left.\begin{matrix} \leq \\ = \\ \geq \end{matrix}\right\} 0, \qquad i = 1, \ldots, m.$$

Equations and inequalities that determine the region of feasibility are usually called constraints or restrictions.

Any *D* satisfying the constraints is known as a feasible solution to the model. The decision-making problem is to find a feasible solution that yields high value or return. An optimal solution (D^*) is defined as a feasible solution producing the greatest possible return, that is,

$$R(Y) = R(D^*, Y) \geq R(D, Y), \qquad D \in S$$
$$= \max_D R(D, Y), \qquad D \in S$$

For every problem, the optimal $R(Y)$ is unique (when it exists) but there may be more than one optimal solution.†

In most real situations it is satisfactory to find a solution yielding a

† The question of existence rarely arises in a real problem but must be explained mathematically. First, if *R* has no upper bound, no maximum is said to exist. A simple example will suffice to explain the second possibility. Suppose $R(D, Y) = D$, where *D* is a real scalar variable restricted to the open interval $0 < D < 1$. The function $R(D, Y)$ has many upper bounds: the smallest is unity. This smallest upper bound is called the *supremum*. Since there is no value of *D* in the open interval, $0 < D < 1$, which yields the supremum, no maximum is said to exist. For almost all problems of practical interest the supremum and maximum are equivalent. We shall only speak of maxima and minima although we could use the more general terms suprema and infima.

near-optimal return. However, the optimal return is established since it is not usually possible to evaluate the goodness of a nonoptimal solution.

The great advantage of a mathematical model is its generality and ease of manipulation. Any sort of sensitivity analysis, such as changing values of the variables, parameters, constraints, or even changing the functional relationships, is most easily accomplished when there is a mathematical model of the system. But these enhanced investigative powers are not attained without cost. The amount of mental effort and analysis required to construct a mathematical model of real-world phenomena is great. First, the system must be described unambiguously. The variables must be identified and a single measure of utility chosen. The relations among the variables must be expressed mathematically. One of these relations is the utility measure and the remainder are constraints.

A solution to a model can be no better than the model itself. Consequently the model must be an accurate representation of the system. But how accurate? Unfortunately, there are no hard and fast rules. The appropriate accuracy of the model depends upon how accurate a solution is needed and how decisions change as the model is modified. Often this can be determined only by trial and error. But it is generally good advice to try the simplest model first. Additional accuracy is likely to mean additional cost and time in constructing the model, and a more accurate model may be difficult to solve and yield no better results.

Various simplifying procedures can be attempted. Decision variables and parameters apparently having negligible effect on the return can be eliminated. The nature of the variables can be changed from continuous to discrete or vice versa. Often in a first model stochastic variations in the parameters are ignored. An obvious simplification is to approximate the return function and constraints by, say, linear functions. We must compromise by balancing solvability and reality. The compromises that must be made will, in part, depend upon the power of the solution techniques. Obviously, sharper optimization tools permit the use of more complicated models.

It is difficult to present a mutually exclusive and collectively exhaustive classification for mathematical models of decision making. But it is useful to make some distinctions. In *deterministic* models the return is given unambiguously by specifying values for the decision variables. There are no uncontrollable or random variables. In contrast, *stochastic* or *probabilistic* models contain random variables that cannot be controlled and whose values are given by probability distributions. A deterministic model can be considered as a special case of a stochastic model, in which each random variable assumes a particular value with probability 1 and all other values with probability 0. In this sense it is possible to treat the two cases together. For the sake of clarity, we shall defer discussion of stochastic models until we have developed the theory and computational aspects concerning deterministic models. A further classification in this direction is *competitive* or *game-theoretic* models, in which different variables are subject to different decision makers' control. We shall elaborate on these classifications in Chapter V.

The continuous or discrete nature of the variables is another mode of classification. This breakdown will be useful in the discussion of computational aspects of dynamic programming in Chapters III and IV. Basically, a continuous variable can assume any real value in an interval, whereas a discrete variable is restricted to a finite number of values in an interval.

Different forms of the objective function and constraints yield still another division of mathematical models. A meaningful grouping is between linear and nonlinear models. In a linear model the objective function is

$$R(D) = c_1 d_1 + c_2 d_2 + \ldots + c_n d_n$$

and the constraints are linear inequalities. Later, we make a very crucial distinction between objective functions of several variables that are separable and those that are not. This technical difference will not be explained now, but will be expanded upon in great detail in Chapter II. An example of a separable function is

$$R(D) = r_1(d_1) + r_2(d_2) + \ldots + r_n(d_n)$$

However, an arbitrary function of n variables is not separable. Closely related to the notion of separability is single-stage versus multistage model. In a single-stage decision process all decisions are made simultaneously, while in a multistage decision process the decisions are made sequentially. This division is not based entirely on the physical characteristics of the process, since it is often possible to create artificially a multistage process from a single-stage one. We would imagine that there are considerable computational advantages to making decisions one at a time rather than all at once. This is the *raison d'être* of dynamic programming.

3. General Approach of Dynamic Programming

Having constructed an appropriate mathematical model, we must choose an optimization technique to solve the model. The way we determine an optimal solution depends, of course, on the form of the objective function and constraints, the nature and number of variables, the kind of computational facilities available, taste, and experience.

Often, before performing the optimization, it is desirable to make some changes of variables and transformations. In contrast to simplifying the model, these preparatory operations preserve the properties of the model completely. The transformed model has the same optimal solution as the original, but is of a form that can be optimized more easily.