Dynamic Programming
in
Chemical Engineering
and
Process Control

SANFORD M. ROBERTS
International Business Machines Corporation
Federal Systems Division
Houston, Texas

1964

New York ACADEMIC PRESS London
To
Nancy, Merrill, and Royce
from
The-Man-in-the-Den
Preface

This book is intended to bridge the gap between dynamic programming as a mathematical discipline and dynamic programming as an engineering tool. It is particularly slanted for the chemical engineer or process engineer whose problems abound in multi-stage decision processes. Dynamic programming is concerned with the general area of multi-stage decision processes. These are processes in which a decision must be made at each stage in such a way that the result of the decision is optimal in light of the entire process rather than in light of the current stage alone. The stages may be stages in time or stages in any sequential operation. Multi-stage decision processes confront us in every avenue of life as well as in mathematics and engineering. The problem of how to spend one's monthly income so that one lives "optimally" between paychecks is a painfully familiar multi-stage decision process.

Throughout the book typical problems are taken as the basis for detailed discussion. The problems chosen are of broad enough scope that the reader should find no difficulty in adapting the approach to his own needs. Sufficient discussion of computational techniques and numerical solutions are given to provide the reader with workable tools.

In Chapter 1 are introduced the basic concepts and the common terminology of dynamic programming. Chapter 2 deals with replacement problems, while Chapter 3 presents a wide range of allocation problems. The very important topic of dynamic programming and the calculus of variations is dealt with in Chapter 4. It is assumed that the reader has or will acquire the rudiments of the calculus of variations which may be found in a number of standard textbooks as background for this chapter. Chapter 5 discusses various aspects of computational techniques so important to dynamic programming. In Chapter 6 we present control problems from the dynamic programming point of view and in Chapter 7 we specialize these controls problems to the field of chemical engineering. Inventory problems, as formulated by dynamic programming, are discussed in Chapter 8. Chapter 9 deals with the challenging field of stochastic problems.
For fulfilling the role of cryptographer-typist the author wishes to thank Kay Dickson, Arlene Kojis, Clydean Norris, Flora Roberts, Nancy Roberts, and Mary Turner. The bulk of the manuscript was typed at Bonner and Moore Associates Inc., while the author was a member of the staff.

The author also wishes to thank for reviewing the text: Richard Bellman and Robert Kalaba of the RAND Corporation, T. M. Stout and W. Brand of TRW Computers Company.

Houston, Texas
August 1963

SANFORD M. ROBERTS
Contents

Preface .................................................. vii

Chapter 1

Introduction

1. Introduction .............................................. 1
2. Definition of Dynamic Programming .................... 2
3. Examples of Multi-Stage Processes ..................... 2
4. Definitions .............................................. 3
5. Method of Attack ....................................... 7
6. Advantages of Dynamic Programming ................... 8
7. Disadvantage of Dynamic Programming ................. 10
Reference ................................................... 10

Chapter 2

Replacement Problems

1. Introduction ............................................. 11
2. Equipment Replacement .................................. 13
3. Numerical Example ..................................... 15
4. Infinitely Long Process Equipment Replacement .......... 17
5. Catalyst Replacement Problems ......................... 19
6. Model 1—Simplest Case .................................. 22
7. Model II—N-Stage Process ............................... 23
8. Computational Aspects for Model II .................... 25
9. Numerical Example, Model II ............................ 27
  References ............................................. 32
  Problems ............................................... 32

Chapter 3

Allocation Problems

1. Introduction ............................................. 39
2. Parallel Reactors, Catalyst Activity Constant .......... 40
3. Parallel Reactors with Feed Forward .................... 41
4. Series Reactor System ................................... 43
Chapter 4

Dynamic Programming and the Calculus of Variations

1. Introduction .................................................. 79
2. Calculus of Variations Problems .......................... 81
3. Ordinary Maxima and Minima and the Calculus of Variations ..... 83
4. Difficulties in the Calculus of Variations .................. 83
5. Equality Constraints ......................................... 84
6. Inequality Constraints ....................................... 85
7. Devices for Handling Inequality Constraints ................ 87
8. Two-Point Boundary Value Problems ....................... 90
9. Linearity ...................................................... 92
10. Nonanalytic Functions ....................................... 93
11. Example: Inequality Constrained Problem ................. 93
12. Dynamic Programming Point of View ..................... 97
14. Max \( \int_0^T f(x, y) \, dt \) .................................. 98
15. Max \( \int_0^T f(x, y) \, dt \) .................................. 102
16. Max \( \int_0^T f(x, y, t) \, dt \) ................................. 104
17. Classical Method for Max \( \int_0^\infty f(x, y) \, dt \) ........... 106
18. Euler-Lagrange Equation Derivation by Dynamic Programming 108
19. Numerical Evaluation of \( f_T = \max_{(\psi)} [F(\psi, v) + G(\psi, v) f_\psi] \) ........ 110
20. Alternate Approach. Finite Analog to Max \( \int_0^T f(x, y) \, dt \) .... 114
21. Finite Analog to Min \( \int_0^T f(y, y', t) \, dt \) ............... 116
22. Control of Chemical Reactor ................................ 117
23. Example: Smoothing Problems as Variational Problems .... 121
24. Example: Brachistochrone ................................ 124
25. Batch Reaction Control, Time Minimization ............... 128
26. Comparison of Methods of Solution of the Brachistochrone Problem 142
27. Dynamic Programming and Calculus of Variations—Summing Up .... 143

References .................................................... 144

Problems ..................................................... 147
Chapter 5

Computational Aspects of Dynamic Programming

1. Introduction ........................................... 157
2. Computational Aspects, Raison d'Être ............... 158
4. General Computation Process ........................... 163
5. Expanding Grid Problem ................................. 166
6. Two-Dimensional Problems ............................... 166
7. Multi-Dimensional Problems ............................. 168
8. Step-by-Step Grid Generation ........................... 169
9. Example of Step-by-Step Grid Generation. Control of Batch Reaction 170
10. Allocation Problems, Dynamic Programming, Differential Calculus 177
11. Successive Approximations, Picard's Method ............ 179
12. Methods of Successive Approximations, Dynamic Programming 181
13. Approximation in Policy Space .......................... 182
14. Approximation in Function Space ......................... 186
15. Confirmation of Previous Examples ....................... 189
16. Quasi-Optimal Policy Approximation ................. 191
17. Alternative Policy Space Approximation ................ 194
18. Analytical Return Functions ............................ 195
19. Lagrangian Multiplier ................................. 197
20. Lagrangian Multiplier Example ........................... 199
21. Lagrangian Multiplier. Counter Example ............... 202
22. Multi-Dimensional Isoperimetric Problem .............. 205
23. The kth Best Policies ................................... 206
24. Numerical Example of the kth Best Policy ............ 209
25. Method of Mechanical Quadrature ....................... 214
26. Mechanical Quadrature and Dynamic Programming ........ 216
27. Reduction in Dimensionality by Linearization. Terminal Control 220
28. Some Properties of Linear Systems ..................... 221
29. Terminal Control for Linearized Equations ............. 222
30. Successive Linear Approximations ....................... 226
31. State of System Specified by Distribution Function .... 227
32. Dynamic Programming and Linear Programming ........ 229
33. Maximization over a Discrete Set ....................... 230
34. Extension to Linear Programming ....................... 231
35. Reduction of Dimensionality in Linear Programming .... 233
36. Lagrangian Multipliers in Linear Programming .......... 235
References .................................................. 237
Problems .................................................. 239

Chapter 6

Control Problems

1. Introduction ........................................... 251
2. Feedback Control and Multi-Stage Decision Process .... 253
3. Averaging Control, General Formulation ................ 253
# Chemical Engineering Control Problems

1. Introduction ........................................ 297
2. Tank Control Problem, Single Tank ............... 298
3. Two-Tank Cascade System .......................... 301
4. Multi-Tank Cascade System ......................... 303
5. Discussion of Cascade Tank Problem .............. 303
6. Terminal Control of a Tubular Reactor ........... 304
7. Averaging Control for Tubular Reactor .......... 311
8. Method of Kalman et al. .......................... 313
9. Linearization around Equilibrium ................. 313
10. Performance Index ................................. 316
11. Control Equations ................................. 317
12. Absorber Example ................................ 323
13. Numerical Evaluation of Absorber Example ...... 326
14. Alternate Method of Kalman ....................... 330
15. Review of Kalman's Method ....................... 332
   References ......................................... 333
   Problems ........................................... 333

# Inventory Problems

1. Introduction ........................................ 343
2. Warehousing Problem ............................... 344
3. Warehousing Problem, Analytical Solution ....... 347
4. Numerical Example, Warehousing Problem ........ 354
5. Comparison of the Warehousing Problem Solutions 355
6. Continuous Inventory Model ....................... 355
Chapter 9

Stochastic Processes

1. Introduction ................................................. 411
2. Deterministic Processes, General Formulation .......... 412
3. Stochastic Process, General Formulation ............... 414
4. Stochastic versus Deterministic Process ................ 416
5. Stochastic Allocation Problem ............................ 418
6. Stochastic Catalyst Regeneration Model ................ 419
7. Stochastic Averaging Control Problem ................... 421
8. Variations on a Theme ..................................... 423
9. Stochastic Matrix in Chemical Engineering ............. 424
10. Conditional Probability, Stochastic Tank Problems .... 425
11. Adaptive Models .......................................... 428
12. Adaptive Averaging Control Model ....................... 430
13. Adaptive Catalyst Replacement Problem ................ 432
    References ................................................ 438
    Problems ................................................. 439

Author Index ................................................... 451
Subject Index .................................................. 453
1. Introduction

In this first chapter, we introduce the reader to the basic concepts and common terminology of dynamic programming. Every idea or term is explained in broad prospective and then highlighted with a "for instance" of some kind. While definitions are notably dull and uninteresting reading, those discussed here form the very basis for a sound comprehension of the subject. Indeed, every concept and term defined in Chapter 1 is used over and over again throughout the text.

To set the groundwork for subsequent chapters, we expose the reader here to three important aspects of dynamic programming: the method of attacking problems by it, the advantages of dynamic programming, and the disadvantages of dynamic programming. For most process engineers, the technique of solving problems by dynamic programming is novel. We attempt in Chapter 1 to help the reader get the "gist" of this new technique. More detailed explanations and examples appear throughout the text.

It is quite natural to ask, when a new method is advocated, what is the contribution of the method above and beyond the conventional way to solve problems. To answer this question, in Section 6 we point out the scope and utility of dynamic programming. In particular, it has the capacity to resolve inequality constrained calculus of variations problems which formerly were ignored or unsolved. Dynamic program-
Dynamic programming can also be used to solve certain combinatorial problems of high dimension.

While extolling the virtues of dynamic programming, it appears worthwhile (and honest) to discuss its drawbacks. Chief among these is the problem of high dimensionality. Another problem is the difficulty in formulating the physical, chemical, control, or economic situation into dynamic programming guise. Both the areas of high dimensionality and problem formulation offer real challenge for creativity and contribution to science and engineering.

2. Definition of Dynamic Programming

Dynamic programming embraces the area of multi-stage decision processes. A multi-stage decision process is one, as the name implies, with many stages involving decisions. In particular, in dynamic programming, the decisions must be made in the light of the entire process rather than in the light of each stage as an entity. This means the decision at each stage must be “right” in view of the entire process.

Dynamic programming is a tool for optimizing mathematical representations of processes. Through it one can find the maximum or minimum of important classes of functions. Since the optimization is over choices of variables over a number of stages, many of the problems that are attacked by dynamic programming might normally be solved by calculus, or in the case of continuous processes, by means of the calculus of variations, which extremizes a quantity involving an integral. The contribution of dynamic programming to problems normally above and beyond the capabilities of the formal calculus of variations is discussed at some length in Chapter 4.

A common theme runs throughout dynamic programming: a criterion of performance to be satisfied, multiplicity of stages in time and/or space, and decisions to be made at each stage to satisfy the objective.

3. Examples of Multi-Stage Processes

In the chemical process industries there are numerous multi-stage decision processes. Typical multi-stage processes are: the allocation of feed in a multi-reactor system, equipment replacement problems, catalyst replacement and regeneration problems, tray-by-tray calculations in distillation and absorption, multiple flash tank separations, and the control of a chemical reaction over time.
Throughout this book many multi-stage optimization processes will be formulated using the techniques of dynamic programming.

4. Definitions

Throughout the application of dynamic programming certain words and ideas occur very frequently. These will be defined here for reference. Better understanding will follow from the examples and from the text as it unfolds.

a. Stage

A stage is a unit in time or space into which the process is divided. A stage may be an arbitrary unit in time, such as a 1-minute or a 1-hour interval, which can be conveniently chosen. A stage may be an equipment unit such as a tray in a distillation column or a reactor in a multi-reactor system. In either case the stage represents a mathematical device for quantizing a continuous variable such as time, or for identifying physical units in a process.

b. State

The state of the system is the set of variables that can be used to describe the system at any stage. In the equipment replacement problem, for example, the state of the system is the age of the equipment. In some problems it may be difficult to clearly specify the state of the system. In one important class of problems the state of the system is given by a distribution function rather than by discrete variables.

c. Functional Equation

In the multi-stage decision process as the process proceeds from stage to stage, the state of the system also changes. The transition from stage to stage and from state to state is described by functional equations. A typical functional equation is

\[ f_N(x) = \max_{0 \leq y_N \leq x} \left( g_N(y_N) + f_{N-1}(x - y_N) \right) \]  

(1)

The notation \( f_N(x) \) is a mathematical shorthand for the following. The term \( f \) refers to an objective, or return, or performance index, perhaps a cumulative profit, to be evaluated by the Principle of Optimality (see Section e). The subscript \( N \) on the \( f \) refers to the number of stages
remaining in the process. The \( x \) in parentheses refers to the state of the system at stage \( N \). In other words, \( f_N(x) \) represents the cumulative value of \( f \) over the \( N \) remaining stages of the process, beginning in state \( x \) and following the Principle of Optimality.

Equation (1) states that the left-hand side of the equation is equal not to the bracketed term on the right-hand side but is equal to the maximum of the bracketed term. Under the Max symbol is a bounded variable, \( y_N \), which designates that the maximization is to be achieved by choosing the proper value for \( y_N \) between the limits 0 to \( x \). The \( y_N \) is the decision variable.

The term \( g_N(y_N) \) represents the gain to be made during stage \( N \) by the optimal choice of \( y_N \) in the region \((0, x)\). If we designate the optimal choice of \( y_N \) as \( y_N^* \), the state of the system is altered from \( x \) in stage \( N \) to \((x - y_N^*) \) in stage \((N - 1)\). The optimal value of the function over the \((N - 1)\) remaining stages is given by \( f_{N-1}(x - y_N^*) \). The term \( f_{N-1}(x - y_N^*) \) represents the cumulative profit over the \((N - 1)\) remaining stages beginning in the state \((x - y_N^*) \). In a similar manner the expression for \( f_{N-1}(x - y_N^*) \) may be evaluated by

\[
f_{N-1}(x - y_N^*) = \max_{0 \leq y_{N-1} \leq x - y_N^*} \left[ g_N(y_{N-1}) + f_{N-2}(x - y_N^* - y_{N-1}) \right]
\]

In Eq. (2) we observe that the choice of \( y_{N-1} \) is restricted to the region \( 0 \) to \( x - y_N^* \) since the original quantity of \( x \) has been reduced to \( x - y_N^* \) due to the allocation of \( y_N^* \) in stage \( N \).

The functional equation (1) is a recursion relationship. It gives a relationship between between an \( N \)-stage process and an \((N - 1)\)-stage process. It also gives a relationship between the states in adjacent stages. The solution of the functional equation yields the value of the return and the corresponding policy, embodied in the set \( \{y_1^*, y_2^*, ..., y_N^*\} \).

d. Policy

A policy is a set of decisions, namely, the values of the \( y_i \) terms, \( i = 1, 2, ..., N \), in Eq. (1) chosen to evaluate the functional equation. In particular, an optimal policy is the set of functions \( y_i \) that maximizes the right-hand side of (1), namely, \( y_1^*, y_2^*, ..., y_N^* \) for \( i = 1, 2, ..., N \).

Often times we are more interested in the optimal policy structure than we are in the value of the optimized function. In general, the value of \( f_N(x) \) is uniquely determined, although there may be a number of optimal policies. This is illustrated, for example, in the numerical results on the control of a batch reaction in Chapter 4, Section 25.
4. DEFINITIONS

e. Principle of Optimality

The "Principle of Optimality" is a cornerstone of dynamic programming. It states "An optimal policy has the property that whatever the initial state and initial decisions are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision."

Utilization of the Principle of Optimality guarantees that the decision made at each stage is the best decision in light of the entire process.

f. Imbedding

Imbedding refers to casting the original problem into a larger class of problems. For example, if the problem we are originally interested in is concerned with only a 10-stage process, when we imbed the original problem we consider an $N$-stage process where $N = 1, 2, \ldots$. The imbedding of the original problem permits us to study and examine an entire class of problems including the original one. From the imbedding process we are able to study both the structure and sensitivity of the solution.

Imbedding amounts to increasing the size and scope of the original problem to gain more information. Like the Lagrange multiplier which also increases the size of the problem, imbedding provides both information and a solution which might not be available otherwise.

Many dynamic programming problems cannot be solved analytically. Instead they are solved by numerical means often requiring computing machinery. The imbedding process surmounts one serious criticism of numerical solution techniques; namely, the lack of generality. The imbedding process in effect provides the generality inherent in analytical solutions by generating a spectrum of solutions.

g. Discrete Process

A discrete process is one in which the variables take on values only at fixed intervals, commonly intervals in time. Between these fixed intervals the variables are not defined. A time period 0 to $T$ may be broken into $N$ increments of size $\Delta$. The time scale reads $1\Delta, 2\Delta, \ldots, N\Delta$, where $N\Delta = T$. A variable such as $x$ is expressed as $x(k\Delta)$ at time $k\Delta$. For convenience $x(k\Delta)$ is often written as $x_k$.

Discrete processes and discrete variables often arise in dynamic programming due to:

(1) finite difference approximations to differential equations,
(2) $N$-stage processes, and
(3) the search for an optimum over a grid of the variables.
h. Continuous Process

A continuous process is one in which the state variables are defined over an interval, commonly an interval of time. We may consider the continuous process to be the limit of the discrete process as the interval between stages becomes small and approaches zero in the limit.

In practice, problems may be formulated as continuous problems, but generally to develop a numerical answer they must be cast into a discrete form for machine computation.

i. Continuous Functional Equation

In dynamic programming it is sometimes useful to employ continuous functional equations rather than discrete functional equations. This is particularly true in establishing the existence and uniqueness of solutions, and in obtaining analytic solutions and approximate solutions. In general, limiting forms are useful for initial study of problems.

Thus, the equation corresponding to Eq. (1) for a process with an infinite number of stages is

$$f(x) = \max_{0 \leq y \leq x} [g(y) + f(x - y)]$$  \hspace{1cm} (3)

j. Successive Approximations

A powerful technique for solving problems is the method of successive approximations. The method consists of assuming a trial value for a variable, substituting the trial value into the equations, and solving. From the computed solution a new trial value is generated and the process repeated until trial and computed values agree to a requisite number of significant figures.

Typical examples of the method of successive approximation are Picard’s method for solving differential equations, and Newton’s method for finding the roots of an equation.

In dynamic programming two methods of successive approximation are employed; one is approximation in policy space, the other is approximation in function space.

k. Approximation in Function Space

Approximation in function space consists in assuming a trial value for the return function $f_N(x)$ or $f(x)$, and then improving upon it. This is the usual approach in successive approximation methods.
5. Method of Attack

1. Approximation in Policy Space

Approximation in policy space is a new contribution offered by dynamic programming. It consists of assuming a trial set of policies for $y_i$ over the $N$-stage process (or over the infinite process) and then improving upon it. In many cases this is a more normal (perhaps a more practically appealing) way to solve the problem, since in practical problems one often has a "feel" for the right thing to do.

Examples of approximation in function space and policy space are given in Chapter 5, Sections 12-17.

5. Method of Attack

In all dynamic programming problems the Principal of Optimality is employed. A particular transliteration of that principle is represented by Eq. (1). Referring to the definition of the Principle of Optimality in Section 4e and Eq. (1) the correspondence is as follows. The initial state is $x$, the initial decision is to choose $y_N = y_N^*$, the remaining decisions are those decisions which generate $f_{N-1}(x - y_N^*)$. By its very definition $f_{N-1}(x - y_N^*)$ is the optimal value of the function $f$ over the $N - 1$ remaining stages beginning in state $(x - y_N^*)$.

In other words, if the "correct" decision is made in stage $N$, all the $(N - 1)$ remaining decisions must be correct as a consequence of the decision in stage $N$. If an "incorrect" choice of $y_N$ is made in stage $N$, then the remaining $(N - 1)$ stages of the process must live with this poor choice. Consistent with this poor choice of $y_N$ for stage $N$, the remaining $(N - 1)$ decisions will be "correct" ones, if the Principle of Optimality is adhered to.

Many of the problems will be cast into the form of the functional equation (1). The first term in the brackets represents the value of the function for stage $N$ and the second term represents the value of the function over all the $(N - 1)$ remaining stages. We may therefore consider the $N$-stage process to be composed of effectively two stages: the current stage and all the remaining stages. The transition from stage to stage and from state to state follows as the value of the $y_i$ term is chosen from stage to stage.

In many dynamic programming problems the equations are solved "backward" so to speak. One starts at the end of the process and moves toward the beginning. The numbering of the stages reflects this by the definition of $N$, which is the number of the stages remaining in the process. For example, $N = 1$ refers to the last stage, one stage remain-
ing; and $N = N$ refers to the beginning of the process when $N$ stages remain.

While in this book the stages are generally counted backward, in some dynamic programming problems the stages are counted forward. In these cases $N = 1$ refers to the initial stage and $N = N$ refers to the final stage. Throughout the text we will specify the counting convention where ever there is any doubt.

The evaluation of the functional equation seldom appears in analytical form. In general, the equation must be evaluated by numerical methods. For example, to start the calculation of the functions $f_N(x)$ appearing in Eq. (1), it is often convenient to first evaluate $f_1(x)$. With $f_1(x)$ known, the function $f_2(x)$ can be found by Eq. (1). With $f_2(x)$ known, $f_3(x)$ can be found by Eq. (1), and so on. The evaluation of first $f_1(x)$, which is used to find $f_2(x)$ which in turn is used to find $f_3(x)$, etc., is ideally suited for machine computation.

Since one does not know the final state of the system, the evaluation of the functional equation (1) is carried out for a series of discrete values for $x$ over the range of interest. In many processes the number of stages to employ is also a problem. Typically one chooses a sufficiently large number of stages so that the operating policy develops a recurring pattern. In the catalyst replacement problem, for example, we are really interested in the catalyst replacement pattern and the profit associated with this cycle, rather than in the profit for the entire $N$-stage process. Once this pattern unfolds, the total number of stages in the process loses its significance.

In the course of generating $f_N(x)$ functions, the decision variables, $y_i$, i.e., the policy, are also determined as a function of $N$. The computational aspects and examples will be given in Chapter 5.

The dynamic programming formulation exhorts the engineer to maximize or minimize. The technique, however, does not spell out how to optimize. It is quite permissible to use any optimizing technique such as differential calculus, gradient methods, search technique, inspection, or even guessing to obtain the solution of equations such as (1) [I].

6. Advantages of Dynamic Programming

If dynamic programming did nothing more than provide another tool to solve problems, which can be solved by other means, it would be valuable in providing an alternate point of view. Dynamic programming, indeed, goes far beyond that worthy point, by providing a new tool to solve problems formerly unsolved or ignored. Dynamic programming can be used to solve many problems in the calculus of variations that
stymie classical techniques. In particular, inequality constraints which are very difficult to handle in the calculus of variations are handled very easily by dynamic programming. For example, for some functions the Euler-Lagrange equations reduce identically to zero, yet a solution exists which can be found by dynamic programming. In many problems, such as species of the brachistochrone problem, dynamic programming solution involves equations which are simpler than the Euler-Lagrange equations. A numerical solution is much easier to find since the differential equations found from dynamic programming are initial value problems while the Euler-Lagrange equations have associated two-point boundary value problems.

Many multi-stage decision processes can be looked upon as combinatorial problems. Consider, for example, an $N$-stage process in which $k$ decisions can be made at each stage (see Fig. 1 for $k = 3, N = 2$).

For each possible decision made in stage $N$, there are $k$ possible decisions in stage $N - 1$. For each possible decision in stage $N - 1$, there are $k$ possible decisions in stage $N - 2$, and so on. A total of $k^N$ possible decision paths must be examined at the beginning of the process to find the optimal solution all at once. While this is possible in principle, in practice it becomes a prohibitively impractical approach even for the fastest computer. Dynamic programming eliminates the need for examining $k^N$ paths at one time by taking each stage as it comes and choosing the best decision out of the $k$ available at each stage. In other words, dynamic programming reduces a combinatorial problem involving $k^N$ choices to a problem requiring only $Nk$ choices, a tremendous reduction in problem size and difficulty.

To illustrate this numerically let us consider first a modest problem where $k = 3$ and $N = 10$. The combinatorial approach requires
examination of $3^{10} \approx 5.9(10^4)$ combinations in contrast to the total of 30 combinations by the step-by-step procedure of dynamic programming. If we consider a process where $k = 3$ and $N = 100$, the combinatorial approach requires inspection of $3^{100} \approx 5.15(10^{47})$ possibilities, while dynamic programming requires inspection of only 300 combinations. The enumeration and ranking of the possibilities for this case by the combinatorial method is a task of some magnitude. If we assume it takes $10^{-6}$ second to evaluate one combinatorial possibility, then approximately $10^{38}$ hours are required. This of course is prohibitively long period of time to wait for an answer.

7. Disadvantages of Dynamic Programming

Setting up problems in dynamic programming is often difficult. As in other fields of mathematics, proper formulation often spells the difference between success and frustration, if not failure. Unlike linear programming where the simplex algorithm is a fairly universal technique, dynamic programming provides no general all-purpose algorithm. Each problem poses its own difficulties and the proper optimizing technique must be sought.

The principal drawback to the use of dynamic programming is the problem of high dimensionality. With many variables to handle, the implementation of a dynamic programming formulation on even the largest computing machines is thwarted by machine memory and speed limitations. To understand this better let us point out that we frequently optimize by search techniques over a grid in the variables. If, we consider, over the region of interest, that ten discrete values are used for each variable, we see that a two-dimensional problem requires 100 nodes to be examined, a three-dimensional problem requires 1000 nodes, etc. If, however, ten discrete values give too coarse a grid, then more discrete values must be chosen, which, of course, only increases the number of nodes to be examined. The larger the number of nodes, the greater the computer memory requirements and the longer the time for computation.

It is extremely important to reduce dimensionality, whenever possible. In Chapter 5 various techniques for reduction of dimensionality are discussed.

REFERENCE

2 Replacement Problems

1. Introduction

In the process industries, the problem of replacing or retiring equipment is very important in terms of plant efficiency, tax considerations, and plant expenses. The equipment replacement problem is part of a larger class of replacement problems that include catalyst replacement, catalyst regeneration, clean up of heat exchangers, and decoking of reactors.

The replacement problems are characterized by the decrease in productivity and efficiency of the equipment over time so that the expense of operating it becomes large enough to warrant replacement.

The approach in the past to replacement problems has been by the calculus, or the calculus of variations, often ignoring complex and realistic aspects. These standard techniques give rise to a set of simultaneous equations that may or may not be readily solved, and, in many cases, do not provide useful information. For industrial problems that often include inequality constraints, due to limits on certain variables or due to discontinuous productivity-cost curves, the standard techniques are often unsatisfactory. In particular, the size of the problem is increased, the equations are more complicated, and sometimes the equations are insoluble without additional techniques.

Dynamic programming, on the other hand, solves problems sequentially so that a set of simultaneous equations is not encountered. In
addition, constraints (inequality or equality) aid rather than hinder the solution of the problem by limiting the solution region.

In Chapter 2, the reader will face his first but not last encounter with the computational aspects of dynamic programming. It will be well for him to carefully study the numerical examples to gain familiarity with the meaning and use of the functional equations. At this juncture, the reader should be able to see for himself how the sequential nature of the dynamic programming solution and how the essentially numerical means of solution can cope with problems that frustrate the calculus and the calculus of variations.

Dynamic programming provides a uniform approach to the class of replacement problems. The formulation consists of two functional equations for each stage of the process. The first equation describes the return for the alternative of keeping the equipment for one more time period. The second alternative describes the return for replacing the equipment. The larger of these two returns determines the course of action chosen, as well as the reward for this choice. As a consequence of the solution of the functional equations, the replacement cycle is determined.

In this chapter, three types of replacement problems are considered. The first problem discussed is an equipment replacement problem where the efficiency of performance at each equipment age is known beforehand. The second is a catalyst replacement problem in which the process is operated in some predetermined but not necessarily optimal manner. At the end of each time period, the question is asked (after the fact) as to whether the catalyst should have been changed. In the third case, a catalyst replacement problem is discussed in which the process variables can be manipulated to adjust the catalyst aging process to maximize profit.

To review the chapter briefly, in Section 2 we describe a simple \( N \) time stage equipment replacement problem. Corresponding to this formulation is a detailed numerical evaluation of the functional equations in Section 3. An extension of the \( N \) time stage process to an infinitely long process is given in Section 4. Section 5 lays the foundation for the catalyst replacement problems. The simplest case is presented in Section 6, while the dynamic programming formulation is given in Section 7. Corresponding to the model in Section 7, the computational aspects and numerical solution are set forth in Sections 8 and 9.

We have chosen fairly simple models for this chapter in order to illustrate the manner of problem formulation and solution. It is quite easy to modify the models to include obsolescence, a price or cost structure that is a function of time, or arbitrary policies such as fixing
2. EQUIPMENT REPLACEMENT

the minimum cycle period. In these more complicated cases, the need for dynamic programming becomes more pronounced.

2. Equipment Replacement

The problem of determining when to replace a piece of equipment depends on the productivity of the equipment, the maintenance cost on the equipment, the trade-in or salvage value of the equipment as a function of the equipment age, and also the purchase cost of a new equipment.

Let us define

\[ r(t) = \text{the value of the productivity per year of a piece of equipment, } t \text{ years old} \]
\[ u(t) = \text{the maintenance cost per year of the equipment, } t \text{ years old} \]
\[ s(t) = \text{the salvage value of the equipment, } t \text{ years old} \]
\[ p = \text{the purchase price of the equipment} \]

Let us now consider a time period of \( N \) years duration in which we desire to determine the optimal replacement cycle. We assume in this process that the functions \( r(t), u(t), \) and \( s(t) \) are known.

We define

\[ f_N(t) = \text{the maximum return from equipment } t \text{ years old over the } N \text{ remaining years of the process upon using an optimal policy} \]

In this process, the age of the equipment is counted forward so that \( t = 0 \) represents a brand new piece of equipment. The time stages of the process are counted backward so that \( N = 1 \) refers to one time stage remaining, and \( N = N \) refers to the beginning of the process (see Fig. 1). While this convention of counting equipment age forward and time stages backward may be somewhat confusing at first, it serves a useful purpose as shown in the functional equations that follow.

\[ Equipment \ Age \]
\[ t \]
\[ Initial \]
\[ N \]
\[ 0 \quad 1 \quad 2 \quad t \]
\[ \text{End} \]
\[ \rightarrow \text{Stages} \]
\[ \text{Fig. 1.} \]

At each stage of the \( N \)-stage process, a decision must be made to keep or to replace the equipment. The decision yielding the higher over-all profit must be accepted.
The functional equations based on the Principle of Optimality are:

\[ f_N(t) = \max \begin{cases} r(t) - u(t) + f_{N-1}(t + 1); \\ s(t) - p + r(0) - u(0) + f_{N-1}(1) \end{cases} \]

Keep

\[ f_1(t) = \max \begin{cases} r(t) - u(t); \\ s(t) - p + r(0) - u(0) \end{cases} \]

Replace

Equation (1) describes the \( N \)-stage process and Eq. (2) the one-stage process. Both equations consist of two parts: the upper equation in each describes the return for the possibility of keeping the equipment; while the lower equation describes the return for replacing the equipment and starting over again with new equipment. In Eq. (1), the term \( r(t) - u(t) \) represents the immediate return of profit less upkeep for stage \( N \) alone. The \( f_{N-1}(t + 1) \) term represents the cumulative profit over the \( (N - 1) \) remaining stages, starting with a piece of equipment that is \( (t + 1) \) years old. The lower line of (1) is interpreted as follows: The function \( s(t) - p \) is the net cost of replacing the equipment of age \( t \) years. The term \( r(0) - u(0) \) is the return on equipment zero years old. It is assumed here that the turning in equipment \( t \) years old for new equipment is done instantaneously, so that the replacement period and the operating period for new equipment take only one stage. The term \( f_{N-1}(1) \) is the return over the \( (N - 1) \) remaining stages starting with equipment one year old.

Similar statements may be made about the one-stage process. In the one-stage process, there is no term such as \( f_0(t + 1) \) because \( N \) runs from 1, 2, ..., \( N \). The relation \( f_0(t) = 0 \) follows from the definition of \( f_N(t) \).

Equations (1) and (2) are recursion expressions that evaluate \( f_N(t) \) in terms of \( f_{N-1}(t + 1) \). In these equations we observe that on passing from one stage to the next the age of the equipment is increased from \( t \) to \( (t + 1) \), and that the number of stages remaining is decreased from \( N \) to \( (N - 1) \); all of which is consistent with our counting convention in Fig. 1. Equation (1) gives a starting point for initiating the computation.

Equations (1) and (2) require that both alternatives of replacing and keeping the equipment be evaluated and the larger return be accepted. These equations yield two pieces of information: the policy decision to keep or replace, and the cumulative profit associated with this decision.

Comment

In this equipment replacement problem our only source of action is to make a choice to keep or replace. We do not have the opportunity
3. **Numerical Example**

In order to fix the ideas and to get a better understanding of the functional equations and their evaluation, we will consider the following numerical example.

Let

\[ r(t) - u(t) = n(t) \]  
\[ s(t) = 0 \]  
\[ p = 10 \]

The values of \( n(t) \) versus \( t \) are given as

\[
\begin{array}{cccccccccccccc}
  t & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \\
n(t) & 10 & 9 & 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 & 0 & 0 \\
\end{array}
\]

Figure 2 is a plot of \( n(t) \) versus \( t \).
Equations (1) and (2) of Section 2 are now written as

\[ f_N(t) = \max \left\{ n(t) + f_{N-1}(t + 1); \quad -p + n(0) + f_{N-1}(1) \right\} \]  

(5)

\[ f_1(t) = \max \left\{ n(t); \quad -p + n(0) \right\} \]  

(6)

The numerical evaluation is as follows.

For a one-stage process and \( t = 0 \), we write

\[ f_1(0) = \max \left\{ n(0); \quad -p + n(0) \right\} = \max \left\{ 10; \quad -10 + 10 = 0 \right\} = 10 \]  

(7)

For a one-stage process and \( t = 1 \), we write

\[ f_1(1) = \max \left\{ n(1); \quad -p + n(0) \right\} = \max \left\{ 9; \quad -10 + 10 = 0 \right\} = 9 \]  

(8)

Continuing in this manner we can develop \( f_1(t) \) which appears as row 1 of Table 1. Note in all cases that in Eq. (6) \( n(t) > -p + n(0) \), so the decision is always to keep the equipment in a one-stage process.

<table>
<thead>
<tr>
<th>( f_N(t) )</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>( f_1(t) )</td>
<td>10</td>
<td>9</td>
<td>8</td>
<td>7</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>( f_2(t) )</td>
<td>19</td>
<td>17</td>
<td>15</td>
<td>13</td>
<td>11</td>
<td>9</td>
<td>9R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( f_3(t) )</td>
<td>27</td>
<td>24</td>
<td>21</td>
<td>18</td>
<td>17</td>
<td>9R</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( f_4(t) )</td>
<td>34</td>
<td>30</td>
<td>26</td>
<td>24</td>
<td>24</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( f_5(t) )</td>
<td>40</td>
<td>35</td>
<td>32</td>
<td>31</td>
<td>30</td>
<td>30R</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( f_6(t) )</td>
<td>45</td>
<td>41</td>
<td>39</td>
<td>37</td>
<td>36</td>
<td>35R</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( f_7(t) )</td>
<td>51</td>
<td>48</td>
<td>45</td>
<td>43</td>
<td>41</td>
<td>41R</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( f_8(t) )</td>
<td>58</td>
<td>54</td>
<td>51</td>
<td>48</td>
<td>48R</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( f_9(t) )</td>
<td>64</td>
<td>60</td>
<td>56</td>
<td>55</td>
<td>54</td>
<td>54R</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( f_{10}(t) )</td>
<td>70</td>
<td>65</td>
<td>63</td>
<td>61</td>
<td>60</td>
<td>60R</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( f_{11}(t) )</td>
<td>75</td>
<td>72</td>
<td>69</td>
<td>67</td>
<td>66</td>
<td>65</td>
<td>65R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( f_{12}(t) )</td>
<td>82</td>
<td>78</td>
<td>75</td>
<td>73</td>
<td></td>
<td>72R</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( f_{13}(t) )</td>
<td>88</td>
<td>84</td>
<td>81</td>
<td>79</td>
<td>78</td>
<td>78R</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( f_{14}(t) )</td>
<td>94</td>
<td>90</td>
<td>87</td>
<td>85</td>
<td>84</td>
<td>84R</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( f_{15}(t) )</td>
<td>100</td>
<td>96</td>
<td>93</td>
<td>91</td>
<td>90</td>
<td>90R</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
For a two-stage process and \( t = 0 \), we write

\[
 f_2(0) = \max \left\{ n(0) + f_1(1); \quad -p + n(0) + f_3(1) \right\} = \max \left\{ 10 + 9 = 19; \quad -10 + 10 + 9 = 9 \right\} = 19 \quad (9)
\]

For a two-stage process and \( t = 1 \), we write

\[
 f_2(1) = \max \left\{ n(1) + f_1(2); \quad -p + n(0) + f_3(1) \right\} = \max \left\{ 9 + 8 = 17; \quad -10 + 10 + 9 = 9 \right\} = 17 \quad (10)
\]

The values for \( f_1(0), f_1(1), f_1(2), \) etc., are found in line 1 of Table 1, which have been developed previously. Continuing in this way we develop \( f_2(t) \), which appears as line 2 of Table 1. From line 2 we generate \( f_3(t) \), which appears as line 3, etc. In Table 1 the "R" appearing after a number identifies the decision choice as replacement and the number gives the return. In generating Table 1 it is assumed that once the decision is made to replace the equipment at a certain return, the return is the same for older equipment. For example, in row 6, at \( t = 6 \), \( f_6(6) = 35 \) and the decision is to replace. For \( f_6(7), f_6(8), f_6(9), \) etc., the return is considered still to be 35.

As marked in Table 1, there are two decision regions. On one side of the line is the decision to replace; on the other is the decision to keep the equipment. From Table 1, the replacement policy can readily be determined. For example, starting with a new piece of equipment in a 15-stage process, the decision is to operate from stage 15 until stage 10 where the decision is to replace. The process is operated from stage 10 to stage 5 where again the equipment must be replaced.

In Chapter 5, the computational aspects of dynamic programming are discussed in more detail. It is important, however, at this point in the text that the reader understand how Table 1 is generated. The working out of this table will do much to aid the reader's comprehension of functional equations.

4. Infinitely Long Process Equipment Replacement

In many processes one would like to look ahead into the far distant future to decide over-all policies. The equipment replacement problem may be looked upon in this light also. One way to do this is to take \( N \) very large and generate the cumulative return and operating policy. Another approach is to develop the equations for an infinite process directly. This we do now.

In order for the infinitely long process not to possess an infinitely large profit, we discount future earnings by the present worth factor, \( a \),
where $0 \leq a < 1$. The "$a$" term may be considered the value of a dollar one year hence.

We define

\[ f(t) = \text{the maximum over-all discounted return from a machine } t \text{ years, using an optimal policy} \]

The counterpart to Eqs. (1) and (2) of Section 2 is

\[
f(t) = \max \begin{cases} r(t) - u(t) + af(t + 1); & \text{Keep} \\ s(t) - p + r(0) - u(0) + af(1); & \text{Replace} \end{cases}
\] (1)

For a new piece of equipment we may show that the policy is always to keep the equipment provided $p \geq s(0)$. At $t = 0$, if the "keep" alternative is larger than the "replace" alternative

\[
\begin{align*}
    r(0) - u(0) + af(1) & \geq s(0) - p + r(0) - u(0) + af(1) \\
    0 & \geq s(0) - p \\
    p & \geq s(0)
\end{align*}
\] (2) (3) (4)

This proves the statement above.

In general for a long period of time, the policy will be to keep the equipment until a replacement time. This means the keep alternative is followed until $T$, the replacement time.

To simplify the algebra, let

\[
n(t) = r(t) - u(t)
\] (5)

Equations (1) may be written as follows:

\[
\begin{align*}
    f(0) &= n(0) + af(1); & \text{Keep} \\
    f(1) &= n(1) + af(2); & \text{Keep} \\
    & \vdots \\
    f(T - 1) &= n(T - 1) + af(T); & \text{Keep} \\
    f(T) &= s(T) - p + n(0) + af(1); & \text{Replace}
\end{align*}
\] (6) (7)

Equations (6) represent the succession of return functions for the "keep" policy. Equation (7) represents the return at time $T$ for the replacement alternative.
Expressing $f(1)$ in terms of the $f(2), f(3),$ and so on by substituting in (6), we obtain

$$f(1) = n(1) + af(2) \quad (8)$$

$$f(1) = n(1) + a[n(2) + af(3)] = n(1) + an(2) + a^2f(3) \quad (9)$$

$$f(1) = n(1) + an(2) + a^2n(3) + \ldots + a^{T-2}n(T-1) + a^{T-1}f(T) \quad (10)$$

Substituting (7) into (10), we find

$$f(1) = n(1) + an(2) + a^2n(3) + \ldots + a^{T-2}n(T-1)$$

$$+ a^{T-3}[s(T) - p + n(0) + af(1)] \quad (11)$$

Collecting terms in $f(1)$, we have

$$f(1) = \frac{n(1) + an(2) + \ldots + a^{T-2}n(T-1) + a^{T-1}[s(T) - p + n(0)]}{1 - a^T} \quad (12)$$

Since $f(1)$ is defined as the maximum over-all return from equipment one year old, Eq. (12) yields the maximum profit. We note that the functions $n(t), s(T),$ and $p$ are known. It remains only to choose the value of $T$ that yields this maximum. The value of $T$ may be found by evaluating (12) for sequentially increasing values of $T$ and comparing to previous values. From the knowledge of $f(1)$ the value $f(0),$ as well as that of $f(t)$ up to $f(T),$ can be found through Eq. (6).

5. Catalyst Replacement Problems

Introduction

In a catalytic reactor process an important objective is to maximize profit by producing on-specification products at the desired rate and by replacing or regenerating the catalyst at the best time. The plant operator desires to run the reactor so that his profit is a maximum over a long period of time. The actions he takes today will influence the behavior of the system for the remainder of the life of the catalyst. Each day the plant operator must decide whether to continue running the plant or to shut down, replace, or regenerate the catalyst. Due to deactivation of the catalyst, and channeling through the catalyst, the best operating conditions change from day to day. The operator must make compensating adjustments for the inherent variability of the process by changing for example, the feed rate, and the operating temperature.
The decision process is clearly a multi-stage one, and, therefore, the problem is amenable to the methodology of dynamic programming.

Statement of the Problem

For a catalytic cracking reaction, we wish to determine, for the entire operating period,

1. the maximum profit and the maximum profit-time function,
2. the temperature, flow rate-time function,
3. the catalyst replacement policy.

We shall discuss two models of increasing complexity. Model I, where the profit-time function is known, considers only item (3), the operating policy. In this model the operator does not knowingly try to adjust process conditions to alter the catalyst life in the long run. He generally follows prescribed company policies. The profit-time relationship is evaluated at the end of each day and a running record is kept. Model II deals with an $N$-stage process, in which the manipulated variables are adjusted deliberately with the view of maximizing the return over the long run. In the course of this the catalyst life is altered to meet the overall profitability.

To illustrate the principles and to be specific we will consider a simple endothermic gas phase reaction in a tubular reactor where compound $A$ cracks to compounds $B$ and $G$.

$$ A \rightarrow B + G $$

The flow sheet, depicted in Fig. 3, consists of the reactor and a distillation train. The unconverted feed material is recycled to the inlet of the reactor.

![Schematic flow sheet](image)
5. CATALYST REPLACEMENT PROBLEMS

We define the following terms:

\[ M = \text{the fresh make-up feed rate of compound A, lb/unit time} \]  
\[ F = \text{the total feed rate of compound A, lb/unit time} \]  
\[ C = \text{the fraction of compound A that reacts to form products B and G} \]  
\[ CF = \text{the products rate of B and G, lb/unit time} \]  
\[ L = (1 - C)F = \text{recycle, the unconverted quantity of compound A, lb/unit time} \]  
\[ Q = \text{the heat transferred to the reactor, BTU/unit time} \]  
\[ T_0 = \text{temperature of the compound A entering the reactor, } ^\circ\text{F} \]  
\[ T = \text{temperature of the reactants leaving the reactor, } ^\circ\text{F} \]

By material balance equations we can write

\[ M = CF \]  
\[ F = M + L \]

Formulation of the Problem

We will assume that the conversion can be expressed through the formula

\[ C = c_1 T - c_2 F - c_3 S \]

where the constants \( c_1 \), \( c_2 \), and \( c_3 \) are evaluated from plant data. The term \( S \) is the cumulative flow rate through the catalyst from a fresh catalyst (where \( S = 0 \)) until the present time. In brief,

\[ S = \sum_{i=1}^{N} F_i \]

where the summation is taken over the life of the catalyst. We shall use \( S \) to characterize the state of the system.

The terms \( c_1 T \) and \( c_2 F \) evaluate the effect of current exit temperature and flow rate on conversion, while the term \( c_3 S \) evaluates the effect of cumulative throughput of the feed material on the degradation of the catalyst.

Although a chemical reactor is a distributed system, expressions such as Eq. (12) can be developed to describe conversion in terms of readily measured quantities such as feed rate and exit temperature. This may be done through a study of the distributed system and/or study of the plant data [4, 8].
2. REPLACEMENT PROBLEMS

The heat balance around the reactor for an endothermic reaction is

\[ Q = F c_v (T - T_o) + HCF \]  
(13)

\[ c_v = \text{the average specific heat of the vapors, BTU/lb °F} \]  
(14)

\[ H = \text{the heat of reaction, BTU/lb} \]  
(15)

In addition to the relationships (11) and (13), the conversion, temperature, and flow rates are constrained by the relations

\[ C_{\text{Min}} \leq C \leq C_{\text{Max}} \]  
(16)

\[ T_{\text{Min}} \leq T \leq T_{\text{Max}} \]  
(17)

\[ F_{\text{Min}} \leq F \leq F_{\text{Max}} \]  
(18)

Constraints are placed on conversion to inhibit the formation of undesirable by-products. The bounds on temperature guarantee that a reaction will take place and that the catalyst will not be ruined by extreme temperatures. The flow rate is constrained due to limitations on feed availability and plant equipment, such as compressors. The temperature and flow rate combination chosen must satisfy Eq. (16).

The profitability per unit time is defined as

\[ P = CFV_1 - MV_2 - QV_3 - LV_4 - V_5 \]  
(19)

where the \( V_1, V_2, V_3, V_4, \) and \( V_5 \) are the values or costs of each item. Included in \( V_1 \) is the combined value of products \( B \) and \( G \). The term \( V_4 \) is the cost of processing the recycle stream through the distillation train. The term \( V_5 \) accounts for fixed charges.

Subject to relationships (11) and (13) and the constraints (16)-(18) we desire to maximize (19) over a period of time including operating plus shutdown time.

6. Model I—Simplest Case

In Model I, the simplest case, a profit-time function is assumed known. Perhaps this has been established by long plant practice. It may or may not be the maximum profit-time function. For this case we desire to know when to shut down the reactor and replace or regenerate the catalyst. The process is depicted in Fig. 4.

We define the following terms:

\[ P(\tau) = \text{the known profit-time function} \]  
(1)
7. Model II—$N$-Stage Process

\[ t^* = \text{the "best" time for shutdown} \] 
\[ s = \text{the number of shutdown days} \] 
\[ R = \text{the cost of shutdown, catalyst replacement, or regeneration} \] 
\[ A(t) = \text{average daily profit} \] 
\[ A(t^*) = \text{maximum average daily profit} \] 

The problem may be formulated and solved by maximizing the average daily profit. The average daily profit is defined as:

\[ A(t) = \frac{\int_0^t P(\tau) \, d\tau - R}{t + s} \] \hspace{1cm} (7)

By differentiating the average daily profit and letting $t = t^*$, we get the maximum average daily profit:

\[ A(t^*) = \frac{\int_0^{t^*} P(\tau) \, d\tau - R}{t^* + s} \] \hspace{1cm} (8)

It is clear, of course, that the maximum average daily profit $A(t^*)$ may be computed simply by evaluating $A(t)$ for various values of $t$ and picking the largest term.

7. Model II—$N$-Stage Process

We desire to determine the temperature and flow-time function in such a manner as to maximize profit over the duration of the process.
We desire also to determine the operating policy of when to replace or regenerate the catalyst. The process is an $N$-stage process where for convenience each stage may be considered to be one day. In this case the flow rate $F$ is the daily flow rate and the $S$ is the cumulative daily flow rate through the catalyst, and $R$ is the cost of shutdown and catalyst replacement or regeneration.

We now define the following sequence of functions:

$$f_N(S) = \text{the maximum profit from an } N\text{-stage process starting with a catalyst in state } S, \text{ following an optimal policy. The quantity } N \text{ refers to the number of stages remaining. For example, } N = 1 \text{ means only one stage remains. The number } N \text{ includes all of the remaining stages, operating stages plus shutdown stages. A stage can be any convenient unit of time. Within an } N\text{-stage process, the reactor may or may not be shut down, and it may be shut down and started up a number of times; } N = 1, 2, 3, \ldots.$$  

$$S = \text{the state of the system characterized by the cumulative throughput of feed material through the reactor up to but not including the stage } N. \text{ In other words, it is the cumulative throughput at the beginning of stage } N. \text{ The term } S \text{ encompasses the range from a newly installed or regenerated catalyst (where } S = 0) \text{ to shutdown.}$$

$$g(S, T, F) = \text{the profit for stage } N \text{ from a system initially in state } S \text{ where } T \text{ and } F \text{ are the values of temperature and flow rate chosen for stage } N$$

$$h(S, F) = S + F = \text{the transformed state of the system, following the selection of } T \text{ and } F \text{ for stage } N$$

In the discussion which follows we use the words “catalyst replacement” and “catalyst regeneration” interchangeably since the equations apply to both situations.

The dynamic programming problem can be formulated as follows using the Principle of Optimality:

$$f_N(S) = \text{Max}_{T, F} \begin{cases} \text{Max}_{T, F} [g(S, T, F) + f_{N-1}(h(S, F))]; \\ [-R + f_{N-1}(0)] \end{cases}$$

$$f_1(S) = \text{Max}_{T, F} [g(S, T, F)]$$

Equations (5) and (6) are subject to Eqs. (16)–(18) of Section 5. The top line on the right-hand side of Eq. (5) evaluates the profit for the case
8. Computational Aspects for Model II

The function $f_N(S)$ is computed for discrete values of $T$ and $F$. In effect, a grid, as coarse or as fine as desired, of $T$ and $F$ is established. Then $f_N(S)$ is determined by a systematic search technique of examining all the nodes of the grid to find the combination of $T$ and $F$ which yield the maximum profit for the $N$-stage process consistent with the constraints on $T$, $F$, and $C$.

The quantities $f_N(S)$ can be evaluated by building up a table (Table 2) starting with $f_1(S)$. Various discrete values of the state of the system $S_i$ are arbitrarily chosen. The procedure is as follows. The quantity $f_1(0)$

<table>
<thead>
<tr>
<th>Number of stages remaining</th>
<th>State of the system</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>$f_1(0)$</td>
</tr>
<tr>
<td>2</td>
<td>$f_2(0)$</td>
</tr>
<tr>
<td>3</td>
<td>$f_3(0)$</td>
</tr>
<tr>
<td>4</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>$N$</td>
<td>$f_N(0)$</td>
</tr>
</tbody>
</table>
(the maximum profit from a one-stage process starting with a system in state $S = 0$) is found by systematically trying all of the grid values of $T$ and $F$ and solving Eqs. (11), (13), and (19) of Section 5 until a maximum profit is attained. The next quantity determined is $f_1(S_1)$, which is the maximum profit from a one-stage process starting with a system in state $S = S_1$. This means that catalyst has been exposed to the cumulative flow $S_1$. For all of the grid values of $T$ and $F$, Eqs. (11), (13), and (19) of Section 5 are evaluated until the maximum profit is attained, which is $f_1(S_1)$. We note that for a one-stage process, the decision is to always continue operating. The terms $f_1(S_i)$ appear as the entries in row 1 of Table 2.

In two-stage and higher-stage processes, the two contending possibilities of continuing to operate or of shutting down the reactor and replacing the catalyst must be evaluated. For a two-stage process beginning with a catalyst in state $S = S_i$, the possibility of continuing to operate is evaluated as follows. For each grid point, $T$ and $F$ are chosen, and the profit for stage 2 only is evaluated by Eqs. (11), (13), and (19) of Section 5. With the $F$ chosen for stage 2, the state of the catalyst for stage 1 is $S = S_i + F$. The maximum profit for stage 1 is, however, tabulated in row 1 of Table 1 for this state. The over-all profit for the two-stage process equals the profit for stage 2 plus stage 1. The calculation is repeated for the various grid values $T$ and $F$ for stage 2, until the over-all profit for a two-stage process is a maximum. This quantity is precisely the value of the top line of Eq. (5) of Section 7. The other possibility of shutting down for a two-stage process beginning with a catalyst in state $S = S_i$ is equal to the cost of shutdown ($-R$) plus the maximum profit from an $(N - 1)$-stage process beginning with a catalyst in state $S = 0$. This is the numerical evaluation of the bottom line of Eq. (5) of Section 7. (This implies that the shutdown period is equal to one stage.) The larger of these two contending possibilities is entered in row 2 of Table 2 as $f_2(S_i)$. In a similar manner the table can be built up by this boot strap operation to give $f_N(S)$.

The calculation of the table yields the catalyst replacement policy. In addition, the temperature, flow rate, conversion, and recycle rate for each stage of the $N$-stage process are evaluated.

In the calculations for $f_N(S)$, the number of trial-and-error calculations for stage $N$ depends directly on the product of the number of flow rates

---

‡Strictly speaking, to guarantee that the value of each possible state is actually given in the table would require a very extensive table. In practice only certain discrete values of $S_i$ can be chosen as table entries. The evaluation of profit for states generated in the course of the solution whose values fall between the chosen states will require a certain amount of interpolation.
9. Numerical Example, Model II

A numerical example of Model II, the \( N \)-stage process, has been worked out using the following values and conditions:

\[
\begin{align*}
c_1 &= 10^{-3} & V_1 &= \$0.10/lb \\
c_2 &= 10^{-5} & V_2 &= \$0.05/lb \\
c_3 &= 10^{-6} & V_3 &= \$3.00/million \text{BTU} \\
c_p &= 0.5 \text{BTU}/\text{lb} \degree\text{F} & V_4 &= \$0.03/lb \\
H &= 300 \text{BTU}/\text{lb} & V_5 &= 0 \\
R &= \$50 \\
5000 &\leq F \leq 10,000 \text{ lb/stage} \\
900 &\leq T \leq 1000 \degree\text{F} \\
C &\leq 0.80
\end{align*}
\]

As an exploratory measure we employed a coarse grid for temperature and flow rate and coarse spacing for the various states of the system. The grid of temperature and flow rates consisted of five temperatures (900, 925, 950, 975, and 1000\degree\text{F}) and six flow rates (5000; 6000; 7000; 8000; 9000; and 10,000). The states of the system considered were in intervals of 10,000. For the states of the system up to about \( S = 400,000 \), the
maximum flow rate of 10,000 lb/stage corresponded to maximum profit. As a result, this coarse interval of 10,000 between states was satisfactory. Had we used smaller intervals between states (such as increments of 2000), the optimum path would have been still to proceed in increments of flow of 10,000 from one stage to another up to about \( S = 400,000 \).

The results of the calculations are given in Tables 3 and 4. In Table 3 the numbers in the rows in the left-hand margin represent the number of stages remaining. The numbers in the columns in the top margin represent \( S \), the various states of the system. The entries in the Table 3 represent the maximum profit for an \( N \)-stage process beginning in state \( S \).

In row 1 of Table 3 are listed the maximum profits for a one-stage process beginning in states \( S = 0; 10,000; 20,000; \) etc. The maximum profit for a two-stage process is listed in row 2 of Table 3. We note that for the two-stage process beginning in state \( S = 0, f_2(0) = \$321 \) for stage 2) + \$313 (for stage 1) = \$634. Similarly, \( f_2(10,000) = \$313 \) (for stage 2) + \$305 (for stage 1) = \$618. The other entries in the table are developed in the same way, as described in Section 8. For this particular example the maximum over-all profit for an \( N \)-stage process requires that the maximum flow rate \( (F = 10,000) \) be used for each stage until the state of the system reaches about 400,000.

The process conditions corresponding to the profit entries in Table 3 are listed in Table 4 for each state of the system. In this particular example the maximum profit for an \( N \)-stage process is equal to the sum of the maximum profits for each stage. We should remember, however, that in general the maximum profit for an \( N \)-stage process is not equal to the sum of the maximum profits for each stage.

As a consequence of the coarse grid of temperature and flow rate, the profit terms listed in Table 3 do not decline monotonically with increasing catalyst age. For example, in row 1 of Table 3 the profit terms starting with state \( S = 0 \) read as follows: 321, 313, 305, 317, 309, 320, etc. From Table 4 we observe that, for the states \( S = 0; 10,000; \) and 20,000, the temperature remains at 900°F and the flow rate at 10,000. At the state \( S = 30,000 \) it is possible to increase the temperature to the next higher value of 925°F and still remain within the conversion constraint. This accounts for the increase in profit from 305 in state \( S = 30,000 \).

If we had used a finer temperature-flow rate grid, the temperatures would have increased in states \( S = 10,000 \) and 20,000 rather than remain at 900°F. In addition, the corresponding profit terms would also have increased. As a consequence of this finer grid, the profit function would decrease monotonically with increasing catalyst age.

In Table 3 we have partitioned by a line the two decision regions of
a

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28

N

10

20

30

40

50

60

70
80

90

100

110

120

I

130

140

I

150

160

State of the system given in units of 1000.

321 313 305 317 309 320 312 304 316 308 319 311 303 295 287 279 271
634 618 622 626 629 632 616 620 624 627 630 614 598 582 566 550 534
939 935 931 946 941 936 932 928 943 938 933 909 885 861 837 813 789
1256 1244 1251 1258 1245 1252 1240 1247 1254 1241 1228 1196 1164 1132 1100 1068 1036
1565 1564 1563 1562 1561 1560 1559 1558 1557 1536 1515 1475 1435 1395 1355 1315 1276
1885 1876 1867 1878 1869 1879 1870 1861 1852 1823 1794 1746 1698 1650 1602 1555 1 1 5 1 5
2197 2180 2183 2186 2188 2190 2173 2156 2139 2102 2065 2009 1953 1897 1842 1183%
2501 2496 2491 2505 2499 2493 2468 2443 2418 2373 2328 2264 2200 2147r
2817 2804 2810 2816 2802 2788 2755 2722 2689 2636 2583 2511 I2451r
3125 3123 3121 3119 3097 3075 3034 2993 2952 2891 2830
3444 3434 3424 3414 3384 3354 3305 3256 3207 3138 3086
3755 3737 3719 3701 3663 3625 3568 3511 3454 3394 3397
4058 4032 4006 3980 3934 3888 3823 3758 3710 3705 3713
4353 4319 4285 4251 4197 4143 4070 4014 4021 4021 4024
4640 4598 4556 4514 4452 4390 4326 4325 4337 4332 4335
4919 4869 4819 4769 4699 4646 4637 4641 4648 4643 4638
5190 5132 5074 5016 4955 4957 4953 4952 4959 4946,494 1=49 17
5453 5387 5321 5272 5266 5273 5264 5263 5262 05249 5236 5212
5708 5634 5577 5583 5582 5584 5575 55662565' 5544 5531 5499
5869 5860 5839 5818 5786
5955 5890 5888 5899 5893 5895 5878 ,
6164 6155 6126 6105 6065
621 1 6201 6204 6210 6204 6198,6181.
6522 6517 6515 6521 6507 6501 6476 6459 6442 6413 6384 6344
6838 6828 6826 6824,6810°6796
6771 6746 6729 6692 6663 6615 6575 6535 6503 6472r
7149 7139 7129 ,
7127 7105 7091 7058 7033 7008 6971 6934 6886 6838 6798 I6788r
7101 I 7099r
7460 7442 7432' 7422 7400 7378 7345 7312 7287 7242 7205 7149 I
7763 /7745/7727 7717 7687 7665 7624 7591 7558 7513 7468 7412 7410r
8066- 8040 8022 8004 7974 7944 7903 7862 7829 7776 7731 7721 77 13r
8361 8335 8309 8291 8253 8223 8174 8133 8092 8039 8040 8024 8016r

0

S"

TABLE 3
MAXIMUM
PROFIT
FOR N-STAGE
PROCESSSTARTING
I N STATE
S

180

~

263 255
518 502
765 742
1005 973
1236 I1206r

170

190

___

W

h)

c
CI
(

r

5

m

z!*
5r

$

i;

E?

c

9


TABLE 4

Process Conditions for Each State at Maximum Profit

<table>
<thead>
<tr>
<th>State:</th>
<th>0</th>
<th>10,000</th>
<th>20,000</th>
<th>30,000</th>
<th>40,000</th>
<th>50,000</th>
<th>60,000</th>
<th>70,000</th>
<th>80,000</th>
<th>90,000</th>
<th>100,000</th>
</tr>
</thead>
<tbody>
<tr>
<td>T (°F):</td>
<td>900</td>
<td>900</td>
<td>900</td>
<td>925</td>
<td>925</td>
<td>950</td>
<td>950</td>
<td>975</td>
<td>975</td>
<td>1000</td>
<td></td>
</tr>
<tr>
<td>F (lb/stage):</td>
<td>10,000</td>
<td>10,000</td>
<td>10,000</td>
<td>10,000</td>
<td>10,000</td>
<td>10,000</td>
<td>10,000</td>
<td>10,000</td>
<td>10,000</td>
<td>10,000</td>
<td>10,000</td>
</tr>
<tr>
<td>C:</td>
<td>0.80</td>
<td>0.79</td>
<td>0.78</td>
<td>0.795</td>
<td>0.785</td>
<td>0.80</td>
<td>0.79</td>
<td>0.78</td>
<td>0.795</td>
<td>0.785</td>
<td>0.80</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>State:</th>
<th>110,000</th>
<th>120,000</th>
<th>130,000</th>
<th>140,000</th>
<th>150,000</th>
<th>160,000</th>
<th>170,000</th>
<th>180,000</th>
<th>190,000</th>
</tr>
</thead>
<tbody>
<tr>
<td>T (°F):</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
</tr>
<tr>
<td>F (lb/stage):</td>
<td>10,000</td>
<td>10,000</td>
<td>10,000</td>
<td>10,000</td>
<td>10,000</td>
<td>10,000</td>
<td>10,000</td>
<td>10,000</td>
<td>10,000</td>
</tr>
<tr>
<td>C:</td>
<td>0.79</td>
<td>0.78</td>
<td>0.77</td>
<td>0.76</td>
<td>0.75</td>
<td>0.74</td>
<td>0.73</td>
<td>0.72</td>
<td>0.71</td>
</tr>
</tbody>
</table>
continuing to operate and shutting down and regenerating. Everything to the right of the line lies in the regenerate region. The value of the maximum profit for regenerating is denoted by an “r” after the number. The profitability for the regeneration decision holds for this state and all higher states. For example, in stage 10, the maximum profit for shutting down and regenerating is $2767 for \( S = 110,000 \) or larger.

The catalyst replacement policy can best be determined by the following example. In a 28-stage process, with the system in state \( S = 120,000 \), the decision is to shut down and regenerate the catalyst for a profit of $8016. The cycle is begun over again with a fresh catalyst. The sequence of profit functions is \( f_{27}(0), f_{26}(10,000), f_{25}(20,000), f_{24}(30,000), \ldots, f_{16}(120,000), \) and \( f_{14}(130,000) \), where we must regenerate again. The corresponding process conditions beginning with \( f_{27}(0) \) are: \( T = 900°F, F = 10,000, C = 0.80; T = 900, F = 10,000, C = 0.79; \) \( T = 900, F = 10,000, C = 0.78; \) \( T = 925, F = 10,000, C = 0.795; \) etc. The path is traced by the diagonal line across Table 3. The catalyst regeneration policy is clearly r ooooooooo o o, where “r” means regenerate the catalyst and “o” means continue to operate.

Discussion

In many catalytic reactions, the product distribution and quantities vary with the energy input level and the residence time in the reactor. The same percentage of conversion of the feed material may yield different and varying amounts of individual products. Under these circumstances another equation must be written for product distribution as a function of conversion, space velocity, and temperature.

Depending on the reaction, the conversion expression may be a function of a variety of variables such as pressure or the impurity level of the feed. The state of the catalyst may be expressed as a function of many variables, perhaps, the products formed. The conversion expression used here is simple enough to illustrate the principles and complicated enough to approximate a real process.

We have discussed here only constraints on \( C, T, \) and \( F \). There may very well be many other constraints, such as requiring the product rate \( CF \) to be a constant. In case of many constraints, Lagrangian multipliers may be employed, using the procedure suggested by Bellman [2] (see Chapter 5, Sections 19-21).

We have discussed here primarily a catalytic reaction, the principles and procedures are readily extended to other chemical reactions. For
example, in a thermal cracking reaction, the reactor would be shut down due to coke and carbon depositing excessively on the tube wall.

REFERENCES


PROBLEMS

1. Referring to the numerical example in Section 3, let us consider that due to inflation the purchase price is projected to increase as follows:

   \[
   t = 0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 10 \ 11 \ 12 \\
   p(t) = 10 \ 10 \ 11 \ 11 \ 12 \ 12 \ 13 \ 13 \ 14 \ 14 \ 15 \ 15 \ 15
   \]

   Evaluate the return and replacement policy. Compare with the results of Section 3.

2. In addition to the conditions in Problem 1, consider the salvage value to vary as follows:

   \[
   t = 0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 10 \ 11 \ 12 \\
   s(t) = 5 \ 4 \ 3 \ 2 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0
   \]

   Evaluate the return and replacement policy. Compare with Problem 1 and the results of Section 3.
3. For the infinitely long equipment replacement problem in Section 4, and for the data in Section 3, find the optimal replacement time $T$ and $f(0)$ for $a = 0.05, 0.10, 0.15,$ and $0.20.$

4. Show that the catalyst replacement problem may be formulated for the infinitely long process as

$$f(S) = \max_{T,F} \left\{ \max_{T,F} g(S, T, F) + af(h(S, F)); \right\}$$

$$-R + af(0)$$

where $a$ is a discount factor, $0 \leq a \leq 1$.

$S =$ Cumulative flow through the catalyst bed
$T =$ Temperature
$F =$ Feed rate
$R =$ Replacement charges

(Roberts, S. M. [7].)

5. Show, by changing our time scale convention so that $N$ and $t$ both are counted forward, that the equipment replacement problem may be formulated as

$$f_N(t) = \max \left\{ r_N(t) - u_N(t) + af_{N+1}(t + 1), \right\} \quad \text{Keep}$$

$$r_N(0) - u_N(0) - c_N(t) + af_{N+1}(t), \quad \text{Replace}$$

where

$f_N(t) =$ the value at year $N$ of the over-all return from a machine $t$ years old, where an optimal replacement policy is employed for the remainder of the process
$r_N(t) =$ the return at year $N$ for a machine $t$ years old
$c_N(t) =$ replacement cost at year $N$ for a machine $t$ years old
$u_N(t) =$ the maintenance cost at year $N$ for a machine $t$ years old
$a =$ discount factor, $0 \leq a \leq 1$

(Dreyfus, S. E. [5].)

6. The formulation in Problem 5 can be used to handle replacement problems in which an improving technology develops.

Let us consider a 10-year process where technology improvement is shown in Table 5. (Note how the revenue increases for the newer machines and the upkeep decreases.)
# 2. Replacement Problems

## Table 5

### Machine made in year 1

<table>
<thead>
<tr>
<th>Age of machine</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>Revenue</td>
<td>90</td>
<td>85</td>
<td>80</td>
<td>75</td>
<td>70</td>
<td>70</td>
<td>60</td>
<td>60</td>
<td>60</td>
<td>60</td>
</tr>
<tr>
<td>Upkeep</td>
<td>20</td>
<td>20</td>
<td>25</td>
<td>25</td>
<td>30</td>
<td>30</td>
<td>35</td>
<td>40</td>
<td>45</td>
<td>50</td>
</tr>
<tr>
<td>Replacement</td>
<td>200</td>
<td>220</td>
<td>240</td>
<td>250</td>
<td>255</td>
<td>260</td>
<td>265</td>
<td>270</td>
<td>270</td>
<td>270</td>
</tr>
</tbody>
</table>

### Machine made in year 2

<table>
<thead>
<tr>
<th>Age of machine</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Revenue</td>
<td>100</td>
<td>90</td>
<td>80</td>
<td>75</td>
<td>70</td>
<td>65</td>
<td>65</td>
<td>65</td>
<td>65</td>
</tr>
<tr>
<td>Upkeep</td>
<td>15</td>
<td>20</td>
<td>20</td>
<td>25</td>
<td>25</td>
<td>30</td>
<td>30</td>
<td>35</td>
<td>35</td>
</tr>
<tr>
<td>Replacement</td>
<td>200</td>
<td>220</td>
<td>240</td>
<td>250</td>
<td>255</td>
<td>260</td>
<td>265</td>
<td>270</td>
<td>270</td>
</tr>
</tbody>
</table>

### Machine made in year 3

<table>
<thead>
<tr>
<th>Age of machine</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Revenue</td>
<td>110</td>
<td>105</td>
<td>100</td>
<td>95</td>
<td>90</td>
<td>80</td>
<td>70</td>
<td>60</td>
</tr>
<tr>
<td>Upkeep</td>
<td>15</td>
<td>15</td>
<td>20</td>
<td>20</td>
<td>25</td>
<td>25</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>Replacement</td>
<td>200</td>
<td>220</td>
<td>240</td>
<td>250</td>
<td>255</td>
<td>260</td>
<td>265</td>
<td>270</td>
</tr>
</tbody>
</table>

### Machine made in year 4

<table>
<thead>
<tr>
<th>Age of machine</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Revenue</td>
<td>115</td>
<td>110</td>
<td>100</td>
<td>90</td>
<td>80</td>
<td>70</td>
<td>60</td>
</tr>
<tr>
<td>Upkeep</td>
<td>15</td>
<td>15</td>
<td>20</td>
<td>20</td>
<td>25</td>
<td>25</td>
<td>30</td>
</tr>
<tr>
<td>Replacement</td>
<td>210</td>
<td>215</td>
<td>220</td>
<td>225</td>
<td>230</td>
<td>235</td>
<td>240</td>
</tr>
</tbody>
</table>

### Machine made in year 5

<table>
<thead>
<tr>
<th>Age of machine</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Revenue</td>
<td>120</td>
<td>115</td>
<td>115</td>
<td>110</td>
<td>105</td>
<td>100</td>
</tr>
<tr>
<td>Upkeep</td>
<td>10</td>
<td>10</td>
<td>15</td>
<td>15</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>Replacement</td>
<td>210</td>
<td>215</td>
<td>220</td>
<td>225</td>
<td>230</td>
<td>235</td>
</tr>
</tbody>
</table>

### Machine made in year 6

<table>
<thead>
<tr>
<th>Age of machine</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Revenue</td>
<td>125</td>
<td>120</td>
<td>110</td>
<td>105</td>
<td>100</td>
<td>135</td>
<td>125</td>
<td>110</td>
<td>105</td>
</tr>
<tr>
<td>Upkeep</td>
<td>10</td>
<td>10</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Replacement</td>
<td>210</td>
<td>220</td>
<td>230</td>
<td>240</td>
<td>250</td>
<td>210</td>
<td>220</td>
<td>230</td>
<td>240</td>
</tr>
</tbody>
</table>

### Machine made in year 7

<table>
<thead>
<tr>
<th>Age of machine</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Revenue</td>
<td>125</td>
<td>120</td>
<td>110</td>
<td>105</td>
</tr>
<tr>
<td>Upkeep</td>
<td>10</td>
<td>10</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>Replacement</td>
<td>210</td>
<td>220</td>
<td>230</td>
<td>240</td>
</tr>
</tbody>
</table>

### Machine made in year 8

<table>
<thead>
<tr>
<th>Age of machine</th>
<th>0</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Revenue</td>
<td>140</td>
<td>135</td>
<td>125</td>
</tr>
<tr>
<td>Upkeep</td>
<td>5</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Replacement</td>
<td>220</td>
<td>230</td>
<td>240</td>
</tr>
</tbody>
</table>

### Machine made in year 9

<table>
<thead>
<tr>
<th>Age of machine</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Revenue</td>
<td>140</td>
<td>135</td>
</tr>
<tr>
<td>Upkeep</td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>Replacement</td>
<td>220</td>
<td>230</td>
</tr>
</tbody>
</table>

### Machine made in year 10

<table>
<thead>
<tr>
<th>Age of machine</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Revenue</td>
<td>140</td>
<td>135</td>
</tr>
<tr>
<td>Upkeep</td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>Replacement</td>
<td>220</td>
<td>230</td>
</tr>
</tbody>
</table>
Let us assume we possess a machine 3 years old with the following characteristics as shown in Table 6.

### TABLE 6
**INCUMBENT MACHINE**

<table>
<thead>
<tr>
<th>Age</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>Revenue</td>
<td>60</td>
<td>60</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>40</td>
<td>40</td>
<td>40</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>Upkeep</td>
<td>55</td>
<td>55</td>
<td>55</td>
<td>60</td>
<td>60</td>
<td>60</td>
<td>60</td>
<td>65</td>
<td>65</td>
<td>70</td>
</tr>
<tr>
<td>Replacement</td>
<td>250</td>
<td>260</td>
<td>270</td>
<td>280</td>
<td>280</td>
<td>290</td>
<td>290</td>
<td>300</td>
<td>300</td>
<td>310</td>
</tr>
</tbody>
</table>

Using the functional equations of Problem 5, develop the return $f_1(3)$ for a 10-year process. Let $a = 1$. *Hint:* Start with $f_{10}(t)$. A machine in year 10, which is $t$ years old, is made in year $(10 - t)$. Check the tabulated results and policy in Table 7. Complete the table delineating the "keep" and the "replacement" boundary regions in the Table 7.

### TABLE 7a

<table>
<thead>
<tr>
<th>$f_N(t)$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_{10}(t)$</td>
<td>130</td>
<td>115</td>
<td>95</td>
<td>85</td>
<td>80</td>
<td>30</td>
<td>30</td>
<td>30</td>
<td>10</td>
<td>40</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$f_9(t)$</td>
<td>240</td>
<td>195</td>
<td>175</td>
<td>165</td>
<td>75</td>
<td>70</td>
<td>60</td>
<td>25</td>
<td>75R</td>
<td>-25R</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$f_8(t)$</td>
<td>310</td>
<td>275</td>
<td>260</td>
<td>145R</td>
<td>125</td>
<td>110R</td>
<td>105R</td>
<td>75</td>
<td>145R</td>
<td>75R</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$f_7(t)$</td>
<td>385</td>
<td>360</td>
<td>215</td>
<td>190</td>
<td>175R</td>
<td>170R</td>
<td>145R</td>
<td>240R</td>
<td>210R</td>
<td>210R</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$f_6(t)$</td>
<td>465</td>
<td>295</td>
<td>265</td>
<td>245R</td>
<td>240R</td>
<td>210R</td>
<td>210R</td>
<td>295R</td>
<td>210R</td>
<td>295R</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$f_5(t)$</td>
<td>435</td>
<td>345</td>
<td>325R</td>
<td>320R</td>
<td>295R</td>
<td>295R</td>
<td>295R</td>
<td>295R</td>
<td>295R</td>
<td>295R</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$f_4(t)$</td>
<td>440</td>
<td>425</td>
<td>280</td>
<td>280</td>
<td>280</td>
<td>280</td>
<td>280</td>
<td>280</td>
<td>280</td>
<td>280</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*The R following a number means that the decision is to replace the equipment of this age.*

7. Over what range of the "$a$" value does the replacement policy in Problem 6 remain unchanged?

8. Let us define $g_N(t, x)$ as the purchase cost of a used machine in year $N$ of age $x$ to replace a machine of age $t$. Show that there are three policies:
to keep the present machine, to purchase a new machine, and to purchase a machine of age $x$:

$$f_N(t) = \text{Max} \begin{cases} r_N(t) - u_N(t) + af_{N+1}(t + 1); \\ r_N(0) - u_N(0) - c_N(t) + af_{N+1}(1); \\ \text{Max}_x [r_N(x) - u_N(x) - g_N(t, x) + af_{N+1}(x + 1)] \end{cases}$$

Keep
Replace with a new machine
Replace with a machine of age $x$

where $f_N(t)$, $c_N(t)$, $u_N(t)$, and $a$ have been defined in Problem 5.

9. We define

$$f_N(t_1, t_2) = \text{the value at year } N \text{ of the over-all return from a machine of age } t_1,$$
and last overhauled at age $t_2$, where an optimal replacement policy is employed for the remainder of the process.

Show that three policies of keeping, purchasing, and overhauling are expressed by the functional equation

$$f_N(t_1, t_2) = \text{Max} \begin{cases} r_N(t_1, t_2) - u_N(t_1, t_2) + af_{N+1}(t_1 + 1, t_2); \\ r_N(0, 0) - u_N(0, 0) - c_N(t_1, t_2) + af_{N+1}(1, 0); \\ r_N(t_1, t_1) - u_N(t_1, t_1) - \theta_N(t_1, t_2) + af_{N+1}(t_1 + 1, t_1) \end{cases}$$

Keep
Replace with new machine
Overhaul

where $\theta_N(t_1, t_2)$ is the cost of overhauling in year $N$ a machine of age $t_1$, last overhauled at age $t_2$.

Parallel definitions hold for $r_N$, the return; $c_N$, the replacement cost; and $u_N$, the maintenance cost.

10. Let us suppose that the quantity of capital outlay for purchasing equipment must be restricted; that is, $c - c_k < M$ where $c = \text{capital on hand}$, $c_k = \text{capital expended at stage } k$, and $M = \text{maximum upper limit}$. Show that the replacement problem may be formulated as

$$f_k(t, c) = \text{Max} \begin{cases} r_k(t) - u_k(t) + af_{k+1}(t + 1, c - u_k(t) + r_k(t)); \\ r_k(0) - u_k(0) - c_k(t) + af_{k+1}(1, c - c_k(t) - u_k(0) + r_k(0)) \end{cases}$$

Keep
Replace

11. If $P(t) = ce^{-t}$, show that the optimal shutdown period time $t^*$ (see Section 6) is found from the solution of

$$(t + s + 1) e^{-t} = 1 - \frac{R}{c}$$

where $R$, $c$, and $s$ are known constants.
12. A thermal cracking tubular reactor cokes up with time and eventually must be cleaned out. We assume the following.
   (a) The outlet pressure is fixed and known, \( a \).
   (b) The inlet pressure varies with time, generally building up to reflect the increasing coking up of the reactor.
   (c) The effective tube diameter may be computed from the over-all pressure drop, flow rate, and cracking temperature.
   (d) The change in inlet pressure at any time is a function of the effective diameter of the previous time period, \( d_e \), the flow rate \( F \), cracking temperature \( T \), and conversion \( C \).

\[
\delta P = \delta P(d_e, T, F, C)
\]

(e) The inlet pressure has an upper limit due to pumping equipment capacity.
(f) Upon cleaning up the reactor, the effective tube diameter is equal to the physical tube diameter and the initial upstream pressure is called \( P_0 \).

Define

\[
f_N(P) = \text{the maximum return from a thermal cracking reactor over the } N \text{ remaining time stages starting with an inlet pressure } P \text{ and following an optimal policy}
\]

\[
g_N(C, T, P, F) = \text{the return from stage } N
\]

\[
R = \text{cost of decoking the thermal cracker}
\]

Show that the decoking cycle is found by

\[
f_N(P) = \text{Max}_{T,F} \left( g_N(C, T, P, F) + f_{N-1}(P + \delta P); \right) \quad \text{Continue}
\]

\[
f_{N-1}(P) = R + f_{N-1}(P_0) \quad \text{Decoke}
\]

where

\[
P \leq P_{\text{Max}}, \quad P_{\text{outlet}} = a
\]

\[
F_{\text{Min}} \leq F \leq F_{\text{Max}}, \quad T_{\text{Min}} \leq T \leq T_{\text{Max}}
\]

\[
C = C(T, F, P, d_e), \quad \delta P = \delta P(d_e, T, F, C)
\]

13. Set up the functional equations for determining the cleaning cycle for a heat exchanger.

14. Set up the functional equations for determining when to back-flow a filter.

15. Set up the functional equations for determining when to reactivat an ion exchanger.

16. Set up the functional equations for determining when to dehumidify a dessicant bed.
17. For a regenerative pebble heat exchanger set up the functional equations to determine the heating cycle.

18. Suppose in the catalyst regeneration problem in Section 7 it takes $M$ days to regenerate the bed. Let

\[ R = \text{total cost of regeneration} \]
\[ f_n(t) = \text{maximum return from a catalyst that was regenerated completely } t \text{ days ago over the } N \text{ remaining stages, if } t \geq 0 \]
\[ f_n(-t) = \text{maximum return from a catalyst with } t \text{ days remaining for complete regeneration over the } N \text{ remaining stages if } t < 0 \]
\[ p(t) = \text{profit from a catalyst that was completely regenerated } t \text{ days ago} \]

Show that the functional equations are

\[
f_n(t) = \text{Max} \begin{cases} 
p(t) + f_{n-1}(t+1), & t \geq 0 \\
-\frac{R}{M} + f_{n-1}(-M+1), & t < 0 
\end{cases}
\]

\[
f_0(t) = \begin{cases} p(t), & t \geq 0 \\
0, & t < 0 
\end{cases}
\]

Note

\[
f_n(-M) = -\frac{R}{M} + f_{n-1}(-M+1) = -\frac{2R}{M} + f_{n-2}(-M+2)
\]
\[
f_n(-M) = -R + f_{n-M}(0)
\]
1. Introduction

An important class of problems that can be attacked by dynamic programming are those involving allocation of a resource. In these problems, a quantity is to be distributed over time or space in order to optimize the over-all return. In fact, any problem involving the sharing or distributing of a resource may be classed as an allocation problem.

In this chapter, we discuss some problems whose allocation aspects are obvious. These include the distribution of a total feed quantity to a number of reactors in parallel or series. There are other problems, however, where the allocation aspect is more recondite. The minimization of the pumping costs in a liquid pipeline is an example. In this problem, the resource to be shared is the pressure boost across all the pumping stations.

In Sections 2—7 a variety of problems dealing with the allocation of feed to multi-reactor systems is discussed. For a multi-reactor system in which each reactor possesses a catalyst of different age, selectivity, and activity, with time invariant properties, Section 2 deals with a parallel system, Section 3 with a parallel system with feed forward, and Section 4 with a series system. Allocation of feed over time to a single reactor with time varying catalyst properties is described in Section 5. An alternate formulation to reduce the state of the system from two to one variable is demonstrated in Section 6. The optimization
of feed allocation over both multiple reactors and time is described in Section 7. One extremely important conclusion to be reached from these various reactor models is that the dynamic programming formulation is independent of the reactor geometry.

A liquid-liquid cross-current extraction process is taken up in Section 8, with an alternate formulation in Section 9. As opposed to the optimization of a single unit, as in Section 8, we deal in Section 10 with the optimization of a multi-unit chemical plant. The minimization of the costs in a liquid pipeline is the subject of Sections 11 and 12. Finally, the maximization of efficiency for units in tandem by means of allocation of costs over the units is presented. In contrast to the usual additive form of the functional equation, we have a product form.

2. Parallel Reactors, Catalyst Activity Constant

We consider here the distribution of a total feed rate \( x \) lb/time to a set of parallel reactors. Each reactor is distinguished by having a catalyst of different age, selectivity, and activity. In this section, we consider the catalyst degradation to be so minor that the catalyst age, selectivity, and activity may be assumed invariant. For simplicity, the only manipulated variable is the feed rate to each reactor. The product rate and quality depend only on the catalyst age, activity, and selectivity and the feed rate. A sketch of the system is given in Fig. 1.

```
    \[
    f_N(x) = \text{the maximum profit obtained by distributing the total flow rate } x \text{ over the } N \text{ reactors following the Principle of Optimality}
    
    g_i(A_i, y_i) = \text{the profit from reactor } i, \text{ of catalyst activity } A_i, \text{ for the flow rate } y_i
    
    \text{and we wish to maximize } \sum_{i=1}^{N} g_i(A_i, y_i) \text{ subject to } \sum_{i=1}^{N} y_i = x.
    
    \text{FIG. 1.}
    
    \text{The state of the system for each reactor depends only on the flow rate, since the catalyst activity is constant.}
    
    \text{We now define}
    
    f_N(x) = \text{the maximum profit obtained by distributing the total flow rate } x \text{ over the } N \text{ reactors following the Principle of Optimality}
    
    g_i(A_i, y_i) = \text{the profit from reactor } i, \text{ of catalyst activity } A_i, \text{ for the flow rate } y_i
    
    \text{and we wish to maximize } \sum_{i=1}^{N} g_i(A_i, y_i) \text{ subject to } \sum_{i=1}^{N} y_i = x.
    
    \text{FIG. 1.}
    
    \text{The state of the system for each reactor depends only on the flow rate, since the catalyst activity is constant.}
    
    \text{We now define}
    ```
The functional equations are

\[ f_N(x) = \max_{0 \leq y_N \leq x} [g_N(A_N, y_N) + f_{N-1}(x - y_N)] \]  

(1)

\[ f_i(x) = \max_{0 \leq y_i \leq x} [g_i(A_i, y_i)] = g_i(A_i, x) \]  

(2)

In Eq. (1), the first term on the right-hand side represents the profit from reactor \( N \) for the choice of \( y = y_N \) where the quantity \( y_N \) is the value of \( y \) in the interval \((0, x)\) which maximizes the expression. The second term represents the profit over the \((N - 1)\) remaining reactors beginning with the quantity of feed \((x - y_N)\). The feed quantity \((x - y_N)\) in turn must be distributed optimally over the \((N - 1)\) remaining reactors. The one-stage reactor profit is described in Eq. (2). The condition \( \Sigma_{i=1}^N y_i = x \) requires that all of the total resource be utilized. The numerical evaluation of this type of equation is discussed at length in Chapter 5.

It is important to recognize in this problem that the optimization is not over time, but over only the reactors themselves.

In the case where there is more than one manipulated variable to generate profit, similar type equations may be set up. If, for example, in the parallel reactor system described above, the temperature in each reactor may be adjusted between upper and lower limits and the flow rate also may be adjusted, the functional equations are:

\[ f_N(x) = \max_{0 \leq y_N \leq x} [g_N(A_N, y_N, T_N) + f_{N-1}(x - y_N)] \]  

(3)

\[ f_i(x) = \max_{0 \leq y_i \leq x} [g_i(A_i, y_i, T_i)] = \max_{T_{\text{Min}} \leq T_i \leq T_{\text{Max}}} [g_i(A_i, x, T_i)] \]  

(4)

where

\( g_i(A_i, y_i, T_i) = \) the profit from reactor \( i \), of catalyst activity \( A_i \), for the flow rate \( y_i \), and the temperature \( T_i \)

3. Parallel Reactors with Feed Forward

An important variation on the feed allocation theme occurs when the unreacted portion of the reactor effluent is fed forward to the next reactor. Again, catalyst activity is different in each reactor and taken to be invariant. Schematically, the system appears in Fig. 2.
We define
\[ C_i = C_i(y_i + r_{i-1}) = \text{the fraction of the total feed rate to reactor } i, \text{ which goes to product; the } C_i \text{ is a function only of the total feed to reactor } i \] (1)

\[ r_i = (1 - C_i)(y_i + r_{i-1}) = \text{the unreacted material in the effluent of the } i\text{th reactor} \] (2)

\[ V_i = \text{the unit value of the product leaving the } i\text{th reactor, } \$/lb; \text{ assume here } V_i \text{ is independent of the conversion and product spectrum} \] (3)

\[ g_i(A_i, y_i + r_{i-1}) = C_i[y_i + r_{i-1}] V_i = \text{the value of the product leaving the } i\text{th reactor} \] (4)

The functional equations are:
\[ f_N(x) = \sum_{0 \leq y_N \leq x} \left[ g_N(A_N, y_N + r_{N-1}) + f_{N-1}(x - y_N) \right] \] (5)

\[ f_i(x) = \sum_{0 \leq y_i \leq x} \left[ g_i(A_i, y_i) \right] = g_i(A_i, x) \] (6)

\[ \sum_{i=1}^{N} y_i = x \] (7)

In these equations, the choice of the \( y_i \) affects not only the profitability in the \( i \)th reactor, but also the profitability in the other reactors through both the remaining fresh feed to be distributed and through the feed
forward from the \((i - 1)\) reactor. For this reason the profit for stage \(N\) is expressed as a function of the total feed to the reactor \(y_N + r_{N-1}\) and \(A_N\), the activity. Note also that after the allocation of the quantity \(y = y_N\) for the \(N\)th reactor, the remaining feed to be allocated is \(x - y_N\), hence the term \(f_{N-1}(x - y_N)\).

4. Series Reactor System

The allocation of a feed resource to a set of reactors in series, as shown schematically in Fig. 3, also may be described. As before, the catalyst activity is considered invariant and different for each reactor. The conversion in each reactor is a function only of the fresh feed material and the unreacted reactor effluent from the preceding reactor:

\[
C_i = C_i(y_i + u_{i+1})
\]  

(1)

The total product yield from an \(N\)-reactor system equals

\[
y_N C_N + [(1 - C_N) y_N + y_{N-1}] C_{N-1} \\
+ [(1 - C_{N-1}) ((1 - C_N) y_N + y_{N-1}) + y_{N-2}] C_{N-2} + ...
\]

(2)

This assumes that the product formed in each reactor is not destroyed or altered in the succeeding reactors.

We let

\[
u_i = \text{the unreacted material leaving reactor } i, \ i = 1, 2, ..., N
\]

(3)

\[
u_{N+1} = 0
\]

\[
u_N = (1 - C_N) y_N
\]

(4)

\[
u_{N-1} = [(1 - C_N) y_N + y_{N-1}] [1 - C_{N-1}]
\]

\[
u_{N-2} = [(1 - C_{N-1}) ((1 - C_N) y_N + y_{N-3}) + y_{N-2}] [1 - C_{N-2}]
\]
The functional equations are:

\[ f_N(x) = \max_{0 \leq t_N \leq x} [g_N(A_N, y_N + u_{N-1}) + f_{N-1}(x - y_N)] \quad (5) \]

\[ f_1(x) = \max_{0 \leq y_1 \leq x} [g(A_1, y_1 + u_2)] = g(A_1, x + u_2) \quad (6) \]

5. Single Reactor, Catalyst Activity Variable

We consider here the problem of distributing a fixed quantity of feed to a single reactor over stages of time in order to maximize profit. The catalyst activity is a variable and catalyst degradation is proportional to the cumulative throughput of fresh feed. The efficacy of the catalyst is measured in terms of conversion and product quality.

For simplicity, the feed rate is the only manipulated variable. The profit depends on the quantity and quality of product produced. The product quality is a function of the current feed rate and the cumulative throughput of feed material through the catalyst bed. The conversion depends on the current feed rate and cumulative feed rate. Simple plots of conversion and quality appear in Figs. 4 and 5.

\[
S_i = \sum_{i=1}^{i} y_i = \text{cumulative feed flow rate at the beginning of time stage } i \quad (1)
\]

\[
C_i = C_i(y_i, S_i) = \text{conversion, fraction of } y_i \text{ that goes to products during } i\text{th time period} \quad (2)
\]

\[
Q_i = Q_i(y_i, S_i) = \text{product quality/unit of } y_i \text{ during the } i\text{th time period} \quad (3)
\]

\[
V_i = V_i(Q_i) = \text{value of product quality per unit of } Q_i \quad (4)
\]
6. Alternate Formulation

An alternate formulation of this problem to reduce the characterization of the state of the system from two terms \( x \) and \( S \) to only one term \( S \) is as follows:

\[
f_N(S) = \max_{0 \leq y_N \leq x} \left[ g_N(y_N, S) - \lambda y_N + f_{N-1}(S + y_N) \right] \tag{1}
\]

\[
f_1(S) = \max_{0 \leq y_1 \leq x} \left[ g_1(y_1, S) - \lambda y_1 \right] = g(x, S) - \lambda x \tag{2}
\]

The \( \lambda \) is a Lagrangian multiplier which is determined numerically so that the cumulative flow rate over the \( N \)-stage time process does not exceed the total cumulative feed resource \( x \). The advantage of reducing the dimensionality or better still the need for reducing the dimensionality whenever possible is discussed in Chapter 5.

7. Feed Rate Distribution over Multiple Reactors and over Time, Variable Catalyst Activity

A much more sophisticated optimization is required when the optimization must be achieved over both space and time. We consider here a parallel multi-reactor system as shown in Fig. 1, in which the catalyst activity and quality vary with the current flow rate and the cumulative feed rate. The feed resource \( x \) must be shared over the \( N \) reactors during the \( M \) periods of time, by proper allocation of the \( y_i \), the flow rate during the \( i \)th time stage through the \( j \)th reactor stage.
Define:

\[ x = \sum_{i=1}^{M} \sum_{j=1}^{N} y_{ij} \]  

(1)

\[ S_{ij} = \sum_{i=1}^{j} y_{ij} \] is the cumulative flow up to the beginning of time stage \( i \) through the \( j \)th reactor

(2)

\[ C_{ij} = C_{ij}(y_{ij}, S_{ij}) \] is the conversion, the fraction of \( y_{ij} \) that goes to products, for the \( i \)th time period and the \( j \)th reactor

(3)

\[ Q_{ij} = Q_{ij}(y_{ij}, S_{ij}) \] is product quality per unit \( y_{ij} \) during the \( i \)th time period for the \( j \)th reactor

(4)

\[ V_{ij} = V_{ij}(Q_{ij}) \] is the value of the product per unit of \( Q_{ij} \)

(5)

The state of the system is specified by the current flow to each reactor and the cumulative flow through each reactor.

For a reactor system of \( N \) parallel reactors and for \( M \) periods of time

\[ f_M(x, S_{M1}, S_{M2}, ..., S_{MN}) \] is the cumulative profit over \( M \) stages of time and \( N \) reactors starting with an initial total resource \( x \), and with the cumulative feed throughput into each reactor \( S_{M1}, S_{M2}, \ldots \), and pursuing an optimal policy

(6)

\[ e_M(y_{M1}, y_{M2}, ..., y_{MN}, S_{M1}, S_{M2}, ..., S_{MN}) = \sum_{j=1}^{N} y_{Mj}C_{Mj}Q_{Mj}V_{Mj} \] is the profit during time stage \( M \) over the \( N \) parallel reactors

(7)

The functional equations are:

\[ f_M[x, S_{M1}, S_{M2}, ..., S_{MN}] = \max_{y_{Mj}} \left[ g_M(y_{M1}, y_{M2}, ..., y_{MN}, S_{M1}, S_{M2}, ..., S_{MN}) + f_{M-1}\left(x - \sum_{j=1}^{N} y_{Mj}, S_{M1} + y_{M1}, S_{M2} + y_{M2}, ..., S_{MN} + y_{MN}\right)\right] \]  

(8)

\[ f_{1}(x, S_{11}, S_{12}, ..., S_{1N}) = \max_{y_{1j}} \left[ g_{1}(y_{11}, y_{12}, ..., y_{1N}, S_{11}, S_{12}, ..., S_{1N}) \right] \]  

(9)

Equation (9) describes a problem seen before, namely, the allocation of feed among reactors of different catalyst activity as if time had no importance. The one-stage process described by Eq. (9) can be evaluated numerically by Eqs. (1) and (2) of Section 2 for each value of \( x \).
A two-stage process described by (8) requires breaking $x$ into the flow rate consumed during time stage 2 plus the flow during time stage 1 and then allocating the flows across all the reactors during each time stage. The execution of Eq. (8) requires a great deal more computation than for a time-invariant problem, although the principles involved are the same. There are, however, considerable computational difficulties when $N > 3$, unless more sophisticated techniques are employed.

Conclusions

The allocation of feed to a variety of multiple reactor systems has been described by dynamic programming. Despite differences in reactor geometry and catalyst properties, the method of dynamic programming provides a universal approach.

8. Cross-Current Extraction

Let us consider the liquid-liquid cross-current extraction system shown in Fig. 6.

We define the following terms:

$Q_o$ = solvent stream rate entering extractor number 1

$W_i$ = wash water rate entering the $i$th extractor

$x_o$ = initial solute composition of $Q_o$

$x_i$ = solute composition of the solvent stream leaving the $i$th extractor

$y_i$ = the solute composition of the wash water stream leaving the $i$th extractor

We make the following assumptions:

(1) the wash water streams $W_i$ and the solvent stream $Q_o$ are immiscible
(2) the wash water streams $W_i$ entering the extractors contain no solute
(3) the effluent streams from each reactor are in equilibrium
3. ALLOCATION PROBLEMS

(4) the total quantity of wash water is known
(5) the value of \( x_0 \) is known
(6) \( Q_0 \) is a known constant.

As a consequence of assumptions (1) and (2) a material balance around the first reactor yields

\[ Q_0(x_0 - x_1) = W_1y_1 \]  

(1)

By the third assumption we have an equilibrium relationship:

\[ y_i = h(x_i) \]  

(2)

Combining (1) and (2) we find that

\[ Q_0(x_0 - x_1) = W_1h(x_1) \]  

(3)

The choice of a value for \( W_1 \) in (3) suffices to determine the \( x_1 \).

In a similar manner by material balance and equilibrium relationships the composition of the solvent stream can be found for any stage of extraction.

An over-all material balance on the solute extracted is given by

\[ \sum_{i=1}^{N} W_i y_i = Q_0(x_0 - x_N) \]  

(4)

By virtue of the fourth assumption we write

\[ \sum_{i=1}^{N} W_i = b \]  

(5)

By the fifth assumption we state

\[ x_0 = a \]  

(6)

Our objective is to maximize the total solute extracted, namely, \( \Sigma_{i=1}^{N} W_i y_i \), by judicious allocation of the wash water streams \( W_i \).

Let us define

\[ f_N(a, b) = \text{the maximum quantity of solute extracted starting with an initial solution concentration } a \text{ and a total wash water } b, \text{ over } N \text{ stages of extraction, and using an optimal policy} \]  

(7)

In this process we will count the stages forward so that \( N = 1 \) refers to the first stage, and \( N = N \) refers to the last stage.
The functional equations are

\[ f_N(a, b) = \max_{0 \leq W_1 \leq b} [Q_0(a - x_1) + f_{N-1}(x_1, b - W_1)] \]  

(8)

\[ f_1(a, b) = \max_{0 \leq W_1 \leq b} [Q_0(a - x_1)] \]  

(9)

In Eq. (8) by the choice of \( W_1 \) through Eqs. (1)-(3), the value of \( x_1 \) is found. The first term in (8) represents the solute extracted in extractor number 1. As a further consequence of the choice of \( W_1 \), the total resource of wash water \( b \) is reduced to \( (b - W_1) \). The second term in (8) represents the total solute extracted over the \( N - 1 \) stages remaining. The first argument in the \( f_{N-1} \) term, \( x_1 \), represents the solute concentration in the solvent entering extractor number 2. The second argument, \( b - W_1 \), represents the wash water available for the remaining \( N - 1 \) stages.

Equation (9) describes a one-extractor stage process.

9. Alternate Formulation of Cross-Current Extraction

In view of the two state arguments \( a \) and \( b \) in Eqs. (7)-(9) of Section 8, we would like to simplify the computations. This may be done once again by the use of a Lagrangian multiplier.

Let us define

\[ f_N(a) = \text{the maximum quantity of solute extracted starting with an initial concentration } x_0 = a, \text{ over } N \text{ stages of extraction, subject to } \sum_{i=1}^{N} W_i = b, \]  

and using an optimal policy

The functional equations are

\[ f_N(a) = \max_{0 \leq W_1 \leq b} [Q_0(a - x_1) - \lambda W_1 + f_{N-1}(x_1, b - W_1)] \]  

(2)

\[ f_1(a) = \max_{0 \leq W_1 \leq b} [Q_0(a - x_1) - \lambda W_1] \]  

(3)

The \( \lambda \), a Lagrangian multiplier, is chosen so that the constraint

\[ \sum_{i=1}^{N} W_i = b \]  

(4)

is satisfied.

To find the proper \( \lambda \) we choose arbitrarily a value for it and solve Eqs. (2) and (3) and test whether (4) is satisfied. If (4) is not satisfied,
another value for $\lambda$ is chosen and the process is repeated until (4) is satisfied.

A common application for the Lagrangian multiplier technique is in a problem with limited resources. We have already seen an example of this in Section 6.

10. Multi-Unit Chemical Plant

A chemical plant consists of a sequence of units made up of reactors, distillation columns, absorbers, and solvent extractors. The effluent from one unit becomes the feed for the succeeding unit. Rather than look at the internal mechanism of each unit we may consider the entire complex as a multi-stage process where we desire to produce products and remove impurities in such a way to maximize profit. In the light of the entire process, we must decide how each unit will perform so that the commodities (quantity and quality) that leave each unit are best. In this problem we are concerned with the gross behavior of the units rather than such details as the temperature profiles in the reactors. For example, we claim the cost or profit of operating a reactor is known at 50, 60, 70% conversion without concerning ourselves with the detailed fuel or steam costs required for the operation of the unit.

Let us consider a general system shown in Fig. 7, which consists of reactors, absorbers, distillation, and extraction units. Feed streams $F_A$ and $F_B$ enter reactor $r$, product streams $F_{1,E}$, $F_{N,E}$, and $F_{N-1,E}$ leave reactors 1, $N$, and $(N - 1)$, respectively.

We define the following terms:

\[ F_{i,i-1} = \text{the flow rate leaving the } i\text{th unit and going to the } (i - 1)\text{th unit} \] (1)

\[ (x_k)_{i,i-1} = \text{the composition of the component } k \text{ in the stream } F_{i,i-1}; \ k = 1, 2, ... \] (2)

\[ (S_k)_{i,i+1} = \text{the separation factor for the } k\text{th component in the effluent stream } F_{i,i+1}, \text{ mols of component } k/\text{total mols entering unit } i; \ k = 1, 2, ... \] (3)

\[ (S_k)_{i,i-1} = \text{the separation factor for the } k\text{th component in the effluent stream } F_{i,i-1}, \text{ mols of component } k/\text{total mols entering unit } i; \ k = 1, 2, ... \] (4)

\[ (v_m)_i = \text{the process variables, such as pressure and temperature, in the } i\text{th unit}; \ m = 1, 2, ... \] (5)

\[ g_i = \text{the profit from the } i\text{th unit; the relative value of the inlet and outlet streams less the operating costs} \] (6)
FIG. 7.
The separation factors for the \( i \)th unit may be evaluated as functions of the entering streams, their compositions, and the process variables:

\[
(S_k)_{i,i+1} = S(F_{i-1,i}, F_{i+1,i}, (x_k)_{i-1,i}, (x_k)_{i+1,i}, (v_1)_i, \ldots, (v_m)_i);
\]

mols of component \( k \) of \( (F_{i-1,i} + F_{i+1,i}) \) in mols \( \quad (7) \)

\[
(S_k)_{i,i+1} = s(F_{i-1,i}, F_{i+1,i}, (x_k)_{i-1,i}, (x_k)_{i+1,i}, (v_1)_i, \ldots, (v_m)_i);
\]

mols of component \( k \) of \( (F_{i-1,i} + F_{i+1,i}) \) in mols \( \quad (8) \)

For the reactor \( r \) with the fresh feed stream of \( F_A \) and \( F_B \), the separation factors appear functionally as

\[
(S_k)_{r,r+1} = H[F_A, F_B, F_{r-1,r}, F_{r+1,r}, (x_k)_A, (x_k)_B, (x_k)_{r-1,r},
(x_k)_{r+1,r}, (v_1)_r, (v_2)_r, \ldots, (v_m)_r];
\]

mols of \( k \) of \( (F_A + F_B + F_{r-1,r} + F_{r+1,r}) \) in mols \( \quad (9) \)

\[
(S_k)_{r,r-1} = h[F_A, F_B, F_{r-1,r}, F_{r+1,r}, (x_k)_A, (x_k)_B, (x_k)_{r-1,r},
(x_k)_{r+1,r}, (v_1)_r, (v_2)_r, \ldots, (v_m)_r];
\]

mols of \( k \) of \( (F_A + F_B + F_{r-1,r} + F_{r+1,r}) \) in mols \( \quad (10) \)

where

\[
(x_k)_A, (x_k)_B = \text{the composition of the } k \text{th component in the streams } F_A \text{ and } F_B,
\]

respectively \( \quad (11) \)

For the reactor \((N - 1)\) with a product stream \( F_C \), the separation factors are expressed functionally:

\[
(S_k)_{N-1,N} = P[F_N, N-1, F_{N-2,N-1}, (x_k)_{N-2,N-1}, (x_k)_{N,N-1},
(v_1)_{N-1}, (v_2)_{N-1}, \ldots, (v_m)_{N-1}];
\]

mols of \( k \) of \( (F_N, N-1 + F_{N-2,N-1}) \) in mols \( \quad (12) \)

\[
(S_k)_{N-1,N-2} = p[F_N, N-1, F_{N-2,N-1}, (x_k)_{N-2,N-1}, (x_k)_{N,N-1},
(v_1)_{N-1}, (v_2)_{N-1}, \ldots, (v_m)_{N-1}];
\]

mols of \( k \) of \( (F_N, N-1 + F_{N-2,N-1}) \) in mols \( \quad (13) \)

\[
(S_k)_{N-1,E} = \Pi [F_N, N-1, F_{N-2,N-1}, (x_k)_{N-2,N-1}, (x_k)_{N,N-1},
(v_1)_{N-1}, (v_2)_{N-1}, \ldots, (v_m)_{N-1}];
\]

mols of \( k \) of \( (F_N, N-1 + F_{N-2,N-1}) \) in mols \( \quad (14) \)
where

\[ (S_k)_{N-1,E} = \text{the separation factor for the product stream } F_c \]  

In a similar manner the separation factors for reactors 1 and \( N \) may be developed.

The separation factors described here are just a convenient way to categorize simply the performance of the distillation, absorption, extraction, and reactor units by a common procedure.

An over-all material balance around the \( i \)th unit yields

\[ F_{i+1,i} + F_{i-1,i} = F_{i,i-1} + F_{i,i+1} \]  

For a system in which no chemical reactions occur, a component material balance may be written as

\[ F_{i+1,i}(x_k)_{i+1,i} + F_{i-1,i}(x_k)_{i-1,i} = F_{i,i-1}(x_k)_{i-1,i} + F_{i,i+1}(x_k)_{i,i+1} \]  

A more general component balance covering chemical reaction systems as well as nonreactive systems may be expressed using the separation factors:

\[ (F_{i+1,i} + F_{i-1,i})(S_k)_{i,i+1} = F_{i,i+1}(x_k)_{i,i+1}; \quad k = 1, 2, ... \]  

\[ (F_{i+1,i} + F_{i-1,i})(S_k)_{i,i-1} = F_{i,i-1}(x_k)_{i,i-1}; \quad k = 1, 2, ... \]  

\[ \sum_{k=1}^{k} F_{i,i+1}(x_k)_{i,i+1} = F_{i,i+1} \]  

\[ \sum_{k=1}^{k} F_{i,i-1}(x_k)_{i,i-1} = F_{i,i-1} \]

Now that the separation factors have been described, we may express the profitability from the \( i \)th unit as

\[ g_i = \sum_{k=1}^{k} [F_{i-1,i} F_{i+1,i} (S_k)_{i,i+1} (S_k)_{i,i-1}] \]  

We desire to maximize the profit over the \( N \) units in Fig. 7; namely, \( \Sigma_{i=1}^{N} g_i \).

The system is constrained by

\[ (F_{i+1,i})_{\text{Min}} \leq (F_{i+1,i})_{\text{Max}}; \quad i = 1, 2, ... \]  

\[ (v_m)_{\text{Min}} \leq v_m \leq (v_m)_{\text{Max}}; \quad m = 1, 2, ... \]  

\[ (F_A)_{\text{Min}} \leq F_A \leq (F_A)_{\text{Max}} \]  

\[ (F_B)_{\text{Min}} \leq F_B \leq (F_B)_{\text{Max}} \]  

\[ (F_C)_{\text{Min}} \leq F_C \leq (F_C)_{\text{Max}} \]
3. ALLOCATION PROBLEMS

We define

\[ f_i(F_{i-1,i}, F_{i+1,i}, (x_i)_{i-1,i}, ..., (x_i)_{i-1,i}, (x_i)_{i+1,i}, ..., (x_i)_{i+1,i}) \]

(28)

the maximum profit over the \( i \) remaining stages, starting with the inlet feed rates and compositions to the \( i \)th unit, and following an optimal policy; the state of the system is specified by the inlet rates and compositions.

The functional equation for the stage one unit is

\[ f_1(F_{2,1}, (x_{1,2,1}), ..., (x_{k,2,1})) = \max_{(v_m)^1} \left[ \sum_{k=1}^{k} g_1(F_{2,1}, (S_k)_{1,E}, (S_k)_{1,2}) \right] \]

(29)

\[ k = 1, 2, ..., m = 1, 2, ... \]

For a two-stage process the functional equation is

\[ f_2(F_{1,2}, F_{3,2}, (x_{1,2,1}), ..., (x_{k,2,1}), (x_{1,3,2}), ..., (x_{k,3,2})) \]

\[ = \max_{(v_m)^2} \left[ \sum_{k=1}^{k} g_2(F_{1,2}, F_{3,2}, (S_k)_{2,3}, (S_k)_{2,1}) + f_1(F_{2,1}, (x_{1,2,1}), ..., (x_{k,2,1})) \right] \]

(30)

\[ k = 1, 2, ..., m = 1, 2, ... \]

In a similar manner the functional equations may be developed for the entire \( N \)-stage process.

The separation factor equations (7)-(14) as well as the material balance equations (17)-(21) are used in conjunction with the functional equations to evaluate the maximum profit.

With the functional equations the maximization may be executed in two ways. The first way considers that there is a known set of separation factors for each unit for each component. The maximization is carried out by the proper choice of the separation factors. No attention is paid to the process conditions required to generate this known set of separation factors. The second method is to choose the process variables in such a way that the profit is maximized over the \( N \) units. The separation factors are evolved as a consequence of the choices for the \( (v_m) \) terms.

Both approaches are valid. The first approach may be used for over-all planning purposes. The second approach may be used for actually controlling the chemical complex.
11. Minimization of Liquid Pipeline Pumping Costs

An interesting application of dynamic programming is found in minimizing pumping costs for a liquid pipeline. Figure 8 is a simplified flow diagram of a series of pumping stations and the corresponding pressure-distance profile.

We assume here that a certain flow rate $Q$ is to be delivered from an initial suction pressure $(P_s)_1$ by a series of pumping stations $L_i$ miles apart to a terminal at a pressure $P_t$. At the booster stations sufficient energy is added to $Q$ to overcome friction losses in the pipeline as well as the elevation heads.

At the $i$th station we may write a pressure balance

$$ (P_d)_i - (P_s)_i = \Delta P_i (L_i+1 - L_i) + K(E_{i+1} - E_i) \quad (1) $$

where

$$ \Delta P_i = \text{friction pressure drop/unit length between station } i \text{ and } i - 1 \quad (2) $$

$$ L_i - L_{i+1} = \text{the length of line between station } i \text{ and } i - 1 \quad (3) $$

$$ E_i - E_{i+1} = \text{the difference in elevation between station } i \text{ and } i - 1 \quad (4) $$

$$ K = \text{proportionality constant to convert elevation head into pressure} \quad (5) $$

The initial suction pressure $(P_s)_1$ and the terminal pressure $P_t$ are known. The optimization is executed by delivering the flow $Q$ so the sum of the energy costs over the $N$ stages (booster stations) is a
minimum. This means finding the proper values for \((P_s)_i\) and \((P_d)_i\) over the \(N\) stages.

We define

\[ y_i = \text{the pressure boost across the } i\text{th pumping station} = (P_d)_i - (P_s)_i \] (6)

\[ x = \sum_{i=1}^{N} y_i = \text{the cumulative pressure boost across the } N\text{ pumping stations} \] (7)

\[ f_N(x) = \text{the minimum cost of delivering } Q \text{ units/time through the } N \text{ pump stations when the sum of the cumulative pressure drop across the } N \text{ stations is } x, \text{ using an optimal policy} \] (8)

\[ g_N(y_N) = \text{the cost of pumping } Q \text{ at station } N \] (9)

We may now write the functional equations

\[ f_N(x) = \min_{0 \leq y_N \leq x} \left[ g_N(y_N) + f_{N-1}(x - y_N) \right] \] (10)

\[ f_1(x) = \min_{0 \leq y_1 \leq x} \left[ g_1(y_1) \right] = g_1(x) \] (11)

In this problem we have at our disposal the allocation of the pressure boost across the \(N\) stations in such a way as to minimize the cost. To be more specific, if we take the cumulative pressure boost across all the pump stations to be \(x = 300\) psi, we determine by dynamic programming how the 300 psi shall be shared among the \(N\) stations.

At each station (each stage) of the process there are constraints on the suction and discharge pressures:

\[ (P_s)_{i_{\text{Min}}} \leq (P_s)_i \leq (P_s)_{i_{\text{Max}}}; \quad i = 1, 2, ..., N \] (12)

\[ (P_d)_{i_{\text{Min}}} \leq (P_d)_i \leq (P_d)_{i_{\text{Max}}}; \quad i - 1, 2, ..., N \] (13)

The execution of Eqs. (10) and (11) must honor the constraints (12) and (13).

The optimization may be carried out in many ways depending on the physical situation in the pipeline. The optimization may be executed by minimizing the cost through employing the proper number and kind of pumps at each station. This requires considering the pump efficiency, the cost of power and utilities, and in many instances the terms of the power contract. For electrically driven pumps the power cost includes a fixed cost for power, whether it is used or not, plus the cost for the energy actually consumed. A further complication in computing power costs is the discontinuities in the energy consumed versus the unit power cost. Typical cost structure is shown in Fig. 9. In some cases
11. MINIMIZATION OF PUMPING COSTS

each pump station has its own electrical power cost curve, such as Fig. 9. The significance of Fig. 9 and the functional equations (10) and (11) is that the cost of a given pressure boost \( y \) across a given pump station is not fixed but is dependent on the cumulative power consumed up to that time.

![Power Cost Schedule](image)

*Total Power Consumed, KVA*

Fig. 9. Power cost schedule.

The minimization also may be affected by considering a variety of impellor sizes for each pump. This amounts to effectively increasing the number of pump combinations available for optimization. The optimization may be carried out in some systems by employing throttling valves which are installed at the discharge of certain stations. To throttle the discharge pressure is, of course, wasteful of power, but this may be the only way in a given physical system to exercise effective control.

For a pipeline of constant diameter where \( \Delta P_i = \Delta P \), Jefferson [6, 7] very cleverly combines the limitations on the suction and discharge pressures to obtain a subsidiary set of inequality constraints which guarantee that the suction and discharge constraints are satisfied.

Following Jefferson, we write

\[
y_i \leq (P_d)_{i_{\text{Max}}} - (P_s)_i \quad (14)
\]

and by (1) and (6)

\[
y_i = \Delta P(L_2 - L_1) + K(E_2 - E_1) \quad (15)
\]

The fact that the suction pressure at stage 2 is at least \((P_s)_{2_{\text{Min}}}\) is given by

\[
(P_d)_1 - \Delta P(L_2 - L_1) - K(E_2 - E_1) \geq (P_s)_{2_{\text{Min}}} \quad (16)
\]
3. ALLOCATION PROBLEMS

Since
\[ y_1 = (P_d)_1 - (P_s)_1 \]  
(17)
it follows that
\[ (P_s)_1 + y_1 - \Delta P(L_2 - L_1) - K(E_2 - E_1) \geq (P_s)_{2_{\text{Min}}} \]  
(18)
\[ y_1 \geq (P_s)_{2_{\text{Min}}} - (P_s)_1 + \Delta P(L_2 - L_1) + K(E_2 - E_1) \]  
(19)
The discharge pressure at station 2 must be less than \((P_d)_{2_{\text{Max}}}:\)
\[ (P_d)_1 - \Delta P(L_2 - L_1) - K(E_2 - E_1) + y_2 \leq (P_d)_{2_{\text{Max}}} \]  
(20)
Using (17), we have
\[ (P_s)_1 + y_1 - \Delta P(L_2 - L_1) - K(E_2 - E_1) + y_2 \leq (P_d)_{2_{\text{Max}}} \]  
(21)
\[ y_1 + y_2 \leq (P_d)_{2_{\text{Max}}} - (P_s)_1 + \Delta P(L_2 - L_1) + K(E_2 - E_1) \]  
(22)
The discharge pressure at the second station must be equal to or greater than \((P_s)_3:\)
\[ (P_s)_1 + y_1 - \Delta P(L_2 - L_1) - K(E_2 - E_1) \]
\[ + y_2 - \Delta P(L_3 - L_2) - K(E_3 - E_2) \geq (P_s)_3 \]  
(23)
or
\[ y_1 + y_2 \geq (P_s)_3 - (P_s)_1 + \Delta P(L_2 - L_1) + K(E_2 - E_1) \]
\[ + \Delta P(L_3 - L_2) + K(E_3 - E_2) \]  
(24)
If we let
\[ B_i = \text{maximum pressure boost across all } i \text{ stations} \]
\[ A_i = \text{minimum pressure boost across all } i \text{ stations} \]
then across station 1 we have
\[ B_1 = (P_d)_{1_{\text{Max}}} - (P_s)_1 \]  
(25)
\[ A_1 = (P_s)_{2_{\text{Min}}} - (P_s)_1 + \Delta P(L_2 - L_1) + K(E_2 - E_1) \]  
(26)
Using (22) and (24), we have across stations 1 and 2:
\[ B_2 = (P_d)_{2_{\text{Max}}} - (P_s)_1 + \Delta P(L_2 - L_1) + K(E_2 - E_1) \]  
(27)
\[ A_2 = (P_s)_{2_{\text{Min}}} - (P_s)_1 + \Delta P(L_2 - L_1) + K(E_2 - E_1) \]
\[ + \Delta P(L_3 - L_2) + K(E_3 - E_2) \]  
(28)
By substituting (25) into (27) and (26) into (28) we may write
\[ B_2 = B_1 + (P_d)_{2_{\text{Max}}} - (P_d)_{1_{\text{Max}}} + \Delta P(L_2 - L_1) + K(E_2 - E_1) \]  
(29)
12. Generalization of Minimized Pumping Costs

A general recursion formula for $A_i$ and $B_i$ emerges from the procedure above:

$$A_i = A_{i-1} - (P_s)_{t_{i-1},\text{Min}} + (P_s)_{t_{i-1},\text{Max}} + \Delta P(L_{i-1} - L_i) + K(E_i - E_{i-1})$$

$$B_i = B_{i-1} - (P_d)_{t_{i-1},\text{Max}} + (P_d)_{t_{i-1},\text{Max}} + \Delta P(L_i - L_{i-1}) + K(E_i - E_{i-1})$$

By using the definitions of $A_i$ and $B_i$, the constraints on the minimum suction pressure and the maximum discharge pressure can be expressed as

$$A_1 \leq y_1 \leq B_1$$
$$A_2 \leq y_1 + y_2 \leq B_2$$
$$\vdots$$
$$A_i \leq y_1 + y_2 + \ldots + y_i \leq B_i$$
$$\vdots$$
$$A_N \leq y_1 + y_2 + \ldots + y_N \leq B_N$$

12. Generalization of Minimization of Liquid Pipeline Pumping Costs

In many pipelines various sections of the line carry different quantities of flow. To be specific, in crude oil lines the quantity of flow generally increases from a minimum at the remotest oil fields to a maximum at the refinery gate, since oil is picked up along the way from other fields. In other instances due to various requirements at the refinery, and due to storage facilities along the pipeline, it is possible to increase or decrease the quantity of crude in a given segment of the line. The pipeline is shown schematically in Fig. 10.

![Fig. 10.](image-url)
In Fig. 10 the flow $Q_i$ is pumped by the $i$th pump station. We assume that due to the storage facilities along the way that all the $Q_i$ terms are different; that is,

$$Q_i \neq Q_{i+1} \quad i = 1, 2, \ldots, N$$

In addition to different flow rates at each pump station, at certain pump stations (but not necessarily all) it may be possible to throttle the discharge pressure. As a result the pressure boost across the station must be sufficient to support the friction drop, the elevation head, and the throttling pressure drop.

Generally, pumps in the stations of a liquid pipeline are connected in series so that each pump handles the total quantity of flow passing through the station. Each pump in effect acts like a stage in a multi-stage pump, where the discharge pressure of the first pump is essentially equal to the suction pressure of the second pump, and so on. Figure 11 illustrates this.

![Cumulative Pressure Boost Across Pump Station](image)

**Fig. 11.** Cumulative pressure boost across pump station.

In view of Fig. 11 it is apparent that in some instances two pumps, say, will deliver almost all the required pressure boost to overcome friction pressure drop and elevation head, but not quite. This means
adding a third pump which supplies not only the necessary head but also “excessive” pressure head. The “excessive” pressure head must be dissipated in one of two ways. It may be dissipated across a throttling valve at the particular station or by others further downstream. The excessive pressure head, on the other hand, may be utilized by the succeeding stations operating at higher suction and discharge datum. Through the application of dynamic programming the proper allocation of pressure boosts across the stations may be determined.

The generalization of the liquid pipeline system in Section 11 to include different quantities of flow in each section of the line and to include throttling capability at some stations may be handled in a manner similar to that in Section 11.

We define

\[ y_i = (P_d)_i - (P_s)_i \] \hspace{1cm} (1)

\[ x = \sum_{i=1}^{N} y_i \] \hspace{1cm} (2)

\[ f_N(x) = \text{The minimum cost of delivering the specified flows } Q_1, Q_2, ..., Q_N \text{ through the pump stations 1, 2, ..., N, respectively, subject to the constraints (7)-(10) and to the pressure balance (11), consuming a cumulative pressure boost } x, \text{ and using an optimal policy} \]

\[ g_N(y_N) = \text{the cost of pumping } Q_N \text{ through station } N \] \hspace{1cm} (4)

The functional equations are

\[ f_N(x) = \min_{0 \leq y_N \leq x} [g_N(y_N) + f_{N-1}(x - y_N)] \] \hspace{1cm} (5)

\[ f_1(x) = \min_{0 \leq y_1 \leq x} [g_1(y_1)] = g_1(x) \] \hspace{1cm} (6)

Equations (5) and (6) are constrained by

\[ (P_s)_{i_{\text{Min}}} \leq (P_s)_i \leq (P_s)_{i_{\text{Max}}} \] \hspace{1cm} (7)

\[ (P_d)_{i_{\text{Min}}} \leq (P_d)_i \leq (P_d)_{i_{\text{Max}}} \] \hspace{1cm} (8)

The flow rates through each pump station are specified as

\[ Q_1 = K_1 \]
\[ Q_2 = K_2 \]
\[ \vdots \]
\[ Q_N = K_N \] \hspace{1cm} (9)
The maximum throttling pressure drop at each station is given by

\[
(\Delta P_i)_1 \leq C_1 \\
(\Delta P_i)_2 \leq C_2 \\
\vdots \\
(\Delta P_i)_N \leq C_N
\]  

(10)

The pressure balance is expressed as

\[
(P_d)_i - (P_s)_i = \Delta P_i(L_{i+1} - L_i) + K(E_{i+1} - E_i) + (\Delta P_i)_i
\]  

(11)

The set of equations (9) covers the range of possibilities from different quantities of flow in each pump station to identical flows in all the stations. The maximum throttling pressure drop described by (10) includes for the \(i\)th station the possibility of no throttling; namely, \(C_i = 0\). The no-throttling condition may come about due to the lack of physical equipment or due to company policy which may forbid throttling at certain stations. Equation (11) merely states that the pressure boost across the \(i\)th station must support the throttling pressure drop as well as the friction flow head and the elevation head.

13. Efficiency of Units in Tandem

In a number of examples given previously the functional equation formulation fell into a form of the type

\[
f_N(x) = \max_{0 \leq y_N \leq x} [g(y_N) + f_{N-1}(x - y_N)]
\]  

(1)

The following example is designed to show that there are other functional equation forms which do appear and which are important in dynamic programming.

Let us consider a number of physical units in series such as a motor, a generator, and a turbine. The output from one serves as the input to the next. Each of these units may be operated at various levels of efficiency. For each level of efficiency there is an associated cost for each unit. We desire to maximize the over-all system efficiency subject to a total over-all fixed cost.
13. EFFICIENCY OF UNITS IN TANDEM

Let us define

\[ C_i = \text{the cost of operating the } i\text{th unit, } C_i > 0 \]  
\[ C = \sum_{i=1}^{N} C_i = \text{the total cost of operating the } N \text{ units} \]  
\[ E_i = \Phi_i(C_i) = \text{the efficiency of the } i\text{th unit, a function of the cost} \]  
\[ f_N(C) = \text{the maximum system efficiency over the } N \text{ units for a total cost} \]  
\[ E = \prod_{i=1}^{N} E_i = \text{the over-all system efficiency} \]

By Eq. (6) we write

\[ E = (E_N) (E_{N-1}) (E_{N-2}) \ldots (E_1) \]  

Substituting (4) into (7), we have

\[ E = [\Phi_N(C_N)] [\Phi_{N-1}(C_{N-1})] [\Phi_{N-2}(C_{N-2})] \ldots [\Phi_1(C_1)] \]  

Using (5) we may express the maximum efficiency at a total cost \( C \):

\[ f_N(C) = \max_{0 \leq C_i \leq C} E = \max_{0 \leq C_i \leq C} \prod_{i=1}^{N} E_i \]  

On substituting (8) into (9) we have

\[ f_N(C) = \max_{0 \leq C_i \leq C} [\Phi_N(C_N)] [\Phi_{N-1}(C_{N-1})] \ldots [\Phi_1(C_1)] \]  

Using the Principle of Optimality, we write the functional equations

\[ f_N(C) = \max_{0 \leq C_i \leq C} [\Phi_N(C_N) f_{N-1}(C - C_N)] \]  

\[ f_1(C) = \max_{0 \leq C_i \leq C} \Phi_1(C_1) \]

We observe that in Eq. (11) the functional equation is expressed as the product of the efficiency for stage \( N \) and the over-all efficiency of the \((N - 1)\) remaining stages. Note the difference between Eqs. (1) and (11).

We may very simply convert the product form of the functional
equation, exemplified by Eq. (11), into the usual additive form. Referring to Eqs. (6)–(8), we define

\[ H = \ln E = \ln [(E_N) (E_{N-1}) \ldots (E_1)] \]  
(13)

\[ H = \ln E = \ln E_N + \ln E_{N-1} + \ldots + \ln E_1 \]  
(14)

\[ H = \ln E = \sum_{i=1}^{N} \ln E_i \]  
(15)

We now define

\[ F_N(C) = \max_{0 \leq C_i \leq C} H = \max_{0 \leq C_i \leq C} \sum_{i=1}^{N} \ln E_i \]  
(16)

Using the Principle of Optimality and Eq. (4), we have

\[ F_N(C) = \max_{0 \leq C_N \leq C} \left[ \ln \Phi_N(C_N) + F_{N-1}(C - C_N) \right] \]  
(17)

\[ F_1(C) = \max_{0 \leq C_1 \leq C} \left[ \ln \Phi_1(C_1) \right] \]  
(18)

By the introduction of logarithms we reverted to the usual additive form upon maximizing the logarithm of the maximum system efficiency.

REFERENCES


PROBLEMS


PROBLEMS

1. For the parallel reactors in Section 2, evaluate \( f_3(5) \) and flow distribution given the following data:

\[
\begin{align*}
  g_1(A_1, y_1) &= C_1 y_1 e^{-A_1} = y_1 e^{-0.1} \\
  g_2(A_2, y_2) &= C_2 y_2^2 e^{-A_2} = \frac{1}{2} y_2^2 e^{-0.2} \\
  g_3(A_3, y_3) &= C_3 y_3^3 e^{-A_3} = \frac{1}{3} y_3^3 e^{-0.3} \\
  \sum_{i=1}^{3} y_i &= 5; \quad y_i = 0, 1, 2, 3, 4, 5
\end{align*}
\]

2. For the 3-reactor system in Section 3, given

\[
\begin{align*}
  C_1 &= \frac{1}{\sqrt{y_1}} e^{-0.05}, \quad g_1 = (\sqrt{y_1} e^{-0.05}) \\
  C_2 &= \frac{1}{\sqrt{y_2 + r_1}} e^{-0.10}, \quad g_2 = (\sqrt{y_2 + r_1} e^{-0.10}) \\
  C_3 &= \frac{1}{\sqrt{y_3 + r_2}} e^{-0.05}, \quad g_3 = (\sqrt{y_3 + r_2} e^{-0.05}) \\
  V_i &= 1 \quad \text{for} \quad i = 1, 2, 3 \\
  \sum_{i=1}^{3} y_i &= 5; \quad y_i = 1, 2, 3, 4, 5
\end{align*}
\]

determine \( f_3(5) \) and the flow distribution.

3. For the series reactor system in Section 4, let

\[
\begin{align*}
  C_1 &= \frac{1}{\sqrt{y_1 + u_2}} e^{-0.05}, \quad g_1 = (\sqrt{y_1 + u_2} e^{-0.05}) \\
  C_2 &= \frac{1}{\sqrt{y_2 + u_3}} e^{-0.10}, \quad g_2 = (\sqrt{y_2 + u_3} e^{-0.10}) \\
  C_3 &= \frac{1}{\sqrt{y_3}} e^{-0.05}, \quad g_3 = \sqrt{y_3} e^{-0.05} \\
  \sum_{i=1}^{3} y_i &= 5; \quad y_i = 1, 2, 3, 4, 5
\end{align*}
\]

Evaluate the return \( f_3(5) \) and the flow distribution.
4. For the system in Section 5, let us minimize the cost so that the equations read

\[ f_N(x, S_N) = \min_{0 \leq y_N \leq x} [g_N(y_N, S_N) + f_{N-1}(x - y_N, S_N + y_N)] \]

\[ f_i(x, S_i) = \min_{0 \leq y_i \leq x} [g_i(y_i, S_i)] = g_i(x, S_i) \]

where

\[ g_N(y_N, S_N) = y_N^2 - S_N + 100 \]

\[ \sum_{i=1}^{4} y_i = 20; \quad y_i = 2, 4, 6, 8, 10 \]

For a 4-stage system, determine \( f_4(20, 0) \) and the flow distribution.

5. Solve Problem 4 using the equivalent equations and the Lagrangian multiplier, namely,

\[ f_N(S) = \min_{0 \leq y_N \leq x} [g_N(y_N, S) - \lambda y_N + f_{N-1}(S + y_N)] \]

\[ f_i(S) = \min_{0 \leq y_i \leq x} [g_i(y_i, S) - \lambda y_i] = g(x, S) - \lambda x \]

6. In the problem in Sections 8 and 9, let us employ a dimensionless flow rate \( v_i \),

\[ v_i = \frac{W_i}{Q_0} \]

since \( Q_0 \) is a constant, and let us show that return function can be written as

\[ g_N(a) = \max_{y_i} \sum_{i=1}^{N} v_i (y_i - \lambda) \]

where \( a = x_0 \) or as

\[ g_N(a) = \max_{y_1} [v_1 (y_1 - \lambda) + g_{N-1}(x_1)] \]

where

\[ x_1 = a - v_1 y_1; \quad g_0(a) = 0 \]

If \( x_1 = H(y_1) \), by the equilibrium relationship, show that \( v_1 \) ranges from zero to \([a - H(\lambda)]/\lambda\).

(Aris, R., Rudd, D. F., and Amundson, N. R. [3].)

7. If the equilibrium relationship is linear, we have for the previous problem

\[ x = \alpha y, \quad \text{where } \alpha \text{ is a known constant} \]
Show that for the one-stage process
\[ g_1(a) = a - (2\alpha \lambda a)^{1/2} + \alpha \lambda \]
where
\[ v_1 = \sqrt{\frac{\alpha a}{\lambda}} - \alpha \quad \text{and} \quad x_1 = \sqrt{\alpha \lambda a} \]
and for the two-stage process
\[ g_2(a) = a - 3(\alpha^2 \lambda^2 a)^{1/3} + 2\alpha \lambda \]
where
\[ v_1 = \sqrt[3]{\frac{\alpha^2 a}{\lambda}} - \alpha \quad \text{and} \quad x_1 = (\alpha \lambda a)^{1/3} \]
\[ v_2 = \sqrt{\frac{\alpha x_1}{\lambda}} - \alpha = \left(\frac{\alpha^2 a}{\lambda}\right)^{1/3} - \alpha = v_1 \]
For the \(N\)-stage process show that
\[ g_N(a) = a - (N + 1) \left(\alpha^N \lambda^N a\right)^{1/N+1} + N\alpha \lambda \]
where
\[ v_1 = \left(\frac{\alpha^N a}{\lambda}\right)^{1/N+1} - \alpha \quad \text{and} \quad x_1 = (\alpha \lambda a)^{1/N+1} \]

8. Show from the previous two examples that the optimal policy is to distribute wash water equally between all stages. By using this fact, show that
\[ \frac{b}{Q_0} = N \left[ \left(\frac{\alpha^N a}{\lambda}\right)^{1/N+1} - \alpha \right] \]
where \(b\) is the total wash water.
Show that
\[ \lambda = \alpha^N a \left[ \frac{b}{NQ_0} + \alpha \right]^{-(N+1)} \]
and as a consequence
\[ g_N(a) = a \left[ 1 - (N + 1) \alpha^N \left(\frac{b}{NQ_0} + \alpha\right)^{-N} + N\alpha^{N+1} \left(\frac{b}{NQ_0} + \alpha\right)^{-(N+1)} \right] \]

9. In a cross-current extraction process, we have a solvent \(B\) which is extracted from a solvent \(A\) by another solvent \(C\). We define the following terms:
\[ x = \text{the fraction of } B/(A + B) \text{ in the raffinate} \]
\[ y = \text{the fraction of } B/(A + B) \text{ in the extract} \]
\[ s = \text{the fraction of } C/(A + B) \text{ in the raffinate} \]
\[ S = \text{the fraction of } C/(A + B) \text{ in the extract} \]
3. ALLOCATION PROBLEMS

\[ q_i = \text{flow rate of } (A + B) \text{ in raffinate leaving } i\text{th stage} \]
\[ u_i = \text{flow rate of } (A + B) \text{ in extract leaving } i\text{th stage} \]
\[ w_i = \text{flow rate of pure solvent } C \text{ entering } i\text{th stage} \]

Show that the material balance for the \( i \)th stage for \((A + B), B, \) and \( C \), respectively, are:

\[
q_{i-1} = q_i + u_i \\
q_{i-1}x_{i-1} = q_ix_i + u_iy_i \\
w_i + q_{i-1}s_{i-1} = q_is_i + u_is_i
\]

If we assume that equilibrium conditions exist at each stage, these equations can be solved.

If we desire to maximize the total amount extracted less the cost of solvent \( C \), show that the equations are given by

\[
g_N(q_0, x_0, s_0) = \max_{w_i} \sum_{i=1}^{N} (u_i y_i - \lambda w_i) \\
g_N(q_0, x_0, s_0) = \max_{w_i} [u_1 y_1 - \lambda w_1 + g_{N-1}(q_1, x_1, s_1)]
\]

By noting that \( s \) and \( x \) are given by the raffinate equilibrium condition at all stages except the first where \( s = 0 \), reduce the formulation to two state variables

\[
g_N(q_0, x_0, 0) = \max_{w_i} [(u_1 y_1 - \lambda w_1) + f_{N-1}(q_1, x_1)]
\]

where

\[
f_N(q_0, x_0) = g_N[q_0, x_0, s(x_0)]
\]

On using the fact that the material balance equations are homogeneous in \( q, w \) and \( u \), we may write

\[
w_i = v_iq_{i-1}; \quad q_i = k_i q_{i-1}; \quad u_i = (1 - k_i) q_{i-1}
\]

Show that the problem can be reduced to one state variable

\[
h_N(x_0) = \max_{v_i} [(1 - k_1) y_1 - \lambda v_1 + k_1 h_{N-1}(x_1)]
\]

\[
h_1(x_0) = \max_{v_i} [(1 - k_1) y_1 - \lambda v_1]
\]

and

\[
f_N(q_0, x_0) = q_0 h_N(x_0) \\
f_1(q_0, x_0) = q_0 h_1(x_0)
\]

(Aris, R., Rudd, D. F., and Amundson, N. R. [3].)
10. It is desired to minimize the total energy expended to compress gas in a multi-stage unit from initial pressure $p$ to final pressure $P$.

The energy $E_N$ is given for the $N$-stage unit by

$$E_N = (nRT) \left( \frac{\gamma}{\gamma - 1} \right) \left[ \left( \frac{P_1}{p} \right)^x + \left( \frac{P_2}{P_1} \right)^x + \ldots + \left( \frac{P}{P_{N-1}} \right)^x - N \right]$$

where

- $P_i$ = the discharge pressure at the $i$th stage
- $n$ = number of mols of gas
- $R$ = universal gas constant
- $T$ = temperature
- $\alpha = (\gamma - 1)/\gamma$
- $\gamma = \frac{C_p}{C_v}$ = rate of specific heats
- $K = (nRT) \left[ \frac{\gamma}{\gamma (\gamma - 1)} \right]$

We minimize $E_N = K \left[ \sum_{i=1}^{N} r_i - N \right]$ subject to

$$r_i \geq 1; \quad i = 1, 2, \ldots, N$$

$$\prod_{i=1}^{N} r_i = r = \frac{P}{p}$$

Define

$$f_k(r) = \text{Min } E_N$$

Show that the functional equations are:

$$f_k(r) = \text{Min } \left[ K(r_k^x - 1) + f_{k-1} \left( \frac{r}{r_k} \right) \right]; \quad k = 2, 3, \ldots, N$$

$$f_1(r) = K(r^x - 1)$$

Show that the optimal policy is for each of the $N$ compressors to do $1/N$ of the work; that is, $r_N = (r)^{1/N}$.

(Aris, R., Bellman, R., and Kalaba, R. [2].)

11. A bank of main heat exchangers, a bank of auxiliary water coolers, and a furnace are shown below. The rate of the main stream to be heated $W$, its temperature $T_N$, and its enthalpy $H_N$ are known, as well as the effluent furnace temperature $T_0$ and enthalpy $H_0$. The cross-stream rates $w_j$, temperature $t_j$, and enthalpy $h_j$, are known. The inlet and outlet cooling water temperatures $\theta_1$ and $\theta_2$ are fixed and identical for all auxiliary exchangers.
For the total heat to be added to the main stream, namely, \(W(H_0 - H_N)\), we desire to minimize the annual charges of all the stages by choosing the proper transfer area for each exchanger and the furnace. The annual charges include the cost of the main and auxiliary heat exchangers, the furnace, the fuel cost for the furnace, and the cooling water costs.

If \(Q\) equals the total heat to be added to the main stream, show that the minimum annual charges \(f_N(Q)\) may be computed by

\[
\frac{\partial f_N(Q)}{\partial Q} = \min_{0 \leq q_N \leq Q} \left[ g_N(q_N) + f_{N-1}(Q - q_N) \right]
\]

where

\[
q_N = \text{the heat added to the main stream in the } N\text{th stage}
\]

\[
g_N(q_N) = \text{the annual cost for the } N\text{th stage of heat exchange including main plus auxiliary exchangers}
\]

\[
s_N = \text{the cooling water rate to the } N\text{th auxiliary exchanger}
\]

Discuss the details of carrying out the evaluation of the functional equations.

12. Solve Problem 11 when the total cooling water utilized, \(S = \sum_{i=1}^{n} s_i\), must be less than \(S_{\text{Max}}\).

13. We desire to maximize \(\sum_{i=1}^{n} x_i\), subject to

\[
x_i \geq 0; \quad \sum_{i=1}^{n} F(x_i) \leq a; \quad \sum_{i=1}^{n} G(x_i) \leq b
\]
If we define

\[ f_n(a, b) = \max \sum_{i=1}^{n} x_i \]

Show that

\[ f_n(a, b) = \max [x_1 + f_{n-1}(a - F(x_1), b - G(x_1))] \]

where \( x_1 \) is bound by

\[ F(x_1) \leq a \quad \text{and} \quad G(x_1) = b \]

or

\[ x_1 \leq F^{-1}(a) \quad \text{and} \quad x_1 = G^{-1}(b) \]

Hence

\[ 0 \leq x \leq \min [F^{-1}(a), G^{-1}(b)] \]

and

\[ f_1(a, b) = \min [F^{-1}(a), G^{-1}(b)] \]

(Bellman, R., Dynamic programming and multi-stage decision process of stochastic type. The RAND Corporation, P-589 (November 2, 1954).)

14. Given

\[ F(x_1, x_2, ..., x_N) = \sum_{i=1}^{N} g_i(x_i) \]

subject to

\[ x_i = 0 \text{ or } 1 \]

\[ \sum_{i=1}^{N} x_i \leq x \]

show that

\[ f_N(x) = \max \sum_{i=0}^{N} [g_N(x_N) + f_{N-1}(x - x_N)] \]

\[ f_N(x) = \max [g_N(1) + f_{N-1}(x - 1), g_N(0) + f_{N-1}(x)] \]

(Bellman, R., Combinatorial processes and dynamic programming. The RAND Corporation, P-1284 (February 24, 1958).)

15. Given that

\[ x_i = 0 \text{ or } 1 \]

\[ \sum_{i=1}^{4} x_i \leq 2 \]

\[ g_1(x_1) = e^{x_1} - 1; \quad g_2(x_2) = e^{-x_2} \]

\[ g_3(x_3) = x_3; \quad g_4(x_4) = \ln(2) - x \]
show that the maximum of $\sum g_i(x_i)$ is 4.411 and that the optimal sequence is
\[ f_4(2), f_3(2), f_2(1), f_1(1) \]

16. Given
\[
\begin{align*}
g_1(x_1) &= a_1 x_1; \quad g_2(x_2) = a_2 x_2^2 \\
g_3(x_3) &= a_3 x_3^3; \quad \ldots; \quad g_N(x_N) = a_N x_N^N \\
x_1 + x_2 + \ldots + x_N &= x \\
x_i &> 0 \\
a_i &> 0
\end{align*}
\]
determine the maximum $\sum_{i=1}^{N} g_i(x_i)$ using the functional equations.

(Bellman, R., Combinatorial processes and dynamic programming. The RAND Corporation, P-1284 (February 24, 1958).)

17. If
\[
\begin{align*}
g_1(x_1) &= 5x_1; \quad g_2(x_2) = 2x_2^2 \\
g_3(x_3) &= \frac{1}{2} x_3^3; \quad g_4(x_4) = \frac{1}{4} x_4^4 \\
x_i &= 1 \text{ or } 2; \quad \sum_{i=1}^{4} x_i \leq 6
\end{align*}
\]
determine the maximum value of $\sum_{i=1}^{4} g_i(x_i)$.

18. Given
\[
\begin{align*}
\sum_{i=1}^{N} x_i &\leq x \\
x_i &= 0, 1 \\
x_i x_{i+1} &= 0; \quad i = 1, 2, \ldots, N - 1 \\
x_N y &= 0 \\
y &= 0, 1
\end{align*}
\]
we desire to maximize $\sum_{i=1}^{N} g_i(x_i)$. Show that
\[
\begin{align*}
f_N(x, y) &= \text{Max} \sum_{i=1}^{N} g_i(x_i) \\
f_N(x, y) &= \text{Max} [g_N(x_N) + f_{N-1}(x - x_N, x_N)]
\end{align*}
\]
and that this leads to two sequences
\[
f_N(x, 0) = \max_0^{y \leq x} [g_N(1) + f_{N-1}(x - 1, 1), g_N(0) + f_{N-1}(x, 0)]
\]
\[
f_N(x, 1) = g_N(0) + f_{N-1}(x, 0)
\]
(Bellman, R., Combinatorial processes and dynamic programming. The RAND Corporation, P-1284 (February 24, 1958).)

19. An entrepreneur has \(x\) dollars available for investment. He can invest immediately in one project and earn a return \(\varphi(y)\) where \(0 \leq y \leq x\). The remaining funds \((x - y)\) increase to \(a(x - y)\) for investment over the \((N - 1)\) remaining stages. Show that the return equation is written as
\[
f_N(x) = \max_{0 \leq y \leq x} [\varphi(y) + f_{N-1}(a(x - y))]
\]
where \(a \geq 1\). If \(\varphi(y) = ky\), show that the return is \(f_N(x) = ka^{N-1}x\) and show that \(y = 0\) for all stages except the last.

20. Referring to the previous problem, if \(\varphi(y) = \sqrt{y}\), show that
\[
f_N(x) = C_N \sqrt{x}
\]
where
\[
C_N = \sqrt{1 + a C_{N-1}^2} \quad \text{and} \quad C_1 = 1; \quad N = 1, 2, 3, \ldots
\]
Show also that the policy is
\[
\frac{y}{x} = \frac{1}{1 + a C_N^2}; \quad N = 1, 2, \ldots
\]
If \(\varphi(y) = y^2\), determine \(f_N(x)\) and the operating policy.

21. Let us maximize \(\sum_{i=1}^{N} g_i(x_i)\) subject to
\[
\sum_{i=1}^{N} x_i \leq x
\]
\[
\sum_{i=1}^{N} a_i x_i \leq y; \quad a_i \geq 0
\]
\[
x_i \geq 0
\]
Show that the functional equations are
\[
f_N(x, y) = \max [g_N(x_N) + f_{N-1}(x - x_N, y - y_N)]
\]
\[
f_i(x, y) = \max [g_i(x_i)]
\]
Show that an alternate formulation is
\[
f_N(x) = \max [g_N(x_N) - \lambda x_N + f_{N-1}(x - x_N)]
\]
(Bellman, R., Combinatorial processes and dynamic programming. The RAND Corporation, P-1284 (February 24, 1958).)
22. Given that
\[ x_i = 0 \text{ or } 1 \]
\[ \sum_{i=1}^{4} x_i \leq x = 2 \]
\[ \sum a_i x_i \leq y = 6 \]
\[ g_1(x_1) = e^{x_1} - 1; \quad g_2(x_2) = e^{-x_2} \]
\[ g_3(x_3) = x_3; \quad g_4(x_4) = \ln(2) - x \]
\[ a_1 = 1, \quad a_2 = 2, \quad a_3 = 3, \quad a_4 = 4 \]
evaluate
\[ f_N(x, y) = \text{Max} [g_N(x_N) + f_{N-1}(x - x_N, y - y_N)] \]
and
\[ f_N(x) = \text{Max} [g_N(x_N) - \lambda x_N + f_{N-1}(x - x_N)] \]

23. In a nuclear power system consisting of a reactor core, turbine, condenser, and pump in series, it is desired to maximize the over-all plant efficiency with respect to cost. Let
\[ C_i = \text{cost of the } i\text{th component} \]
\[ \phi_i(C_i) = \text{cost efficiency of the } i\text{th component} \]
\[ E_i = \text{efficiency of } i\text{th unit} \]
\[ E = \Pi E_i = \text{over-all system efficiency} \]

Table 1 gives the efficiency and cost functions \( E_R \), \( C_R \) for the reactor, \( E_T \), \( C_T \) for the turbine, \( E_C \), \( C_C \) for the condenser, and \( E_P \), \( C_P \) for the pump.

**TABLE 1**

<table>
<thead>
<tr>
<th>( E_R )</th>
<th>( C_R )</th>
<th>( E_T )</th>
<th>( C_T )</th>
<th>( E_C )</th>
<th>( C_C )</th>
<th>( E_P )</th>
<th>( C_P )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.2</td>
<td>10</td>
<td>0.8</td>
<td>10</td>
<td>0.6</td>
<td>10</td>
<td>0.7</td>
<td>10</td>
</tr>
<tr>
<td>0.3</td>
<td>9</td>
<td>0.8</td>
<td>9</td>
<td>0.5</td>
<td>9</td>
<td>0.8</td>
<td>9</td>
</tr>
<tr>
<td>0.4</td>
<td>8</td>
<td>0.8</td>
<td>8</td>
<td>0.4</td>
<td>8</td>
<td>0.7</td>
<td>8</td>
</tr>
<tr>
<td>0.5</td>
<td>7</td>
<td>0.7</td>
<td>7</td>
<td>0.3</td>
<td>7</td>
<td>0.7</td>
<td>7</td>
</tr>
<tr>
<td>0.6</td>
<td>6</td>
<td>0.6</td>
<td>6</td>
<td>0.2</td>
<td>6</td>
<td>0.6</td>
<td>6</td>
</tr>
<tr>
<td>0.5</td>
<td>5</td>
<td>0.5</td>
<td>5</td>
<td>0.2</td>
<td>5</td>
<td>0.5</td>
<td>5</td>
</tr>
<tr>
<td>0.4</td>
<td>4</td>
<td>0.4</td>
<td>4</td>
<td>0.2</td>
<td>4</td>
<td>0.5</td>
<td>4</td>
</tr>
<tr>
<td>0.3</td>
<td>3</td>
<td>0.2</td>
<td>3</td>
<td>0.2</td>
<td>3</td>
<td>0.4</td>
<td>3</td>
</tr>
<tr>
<td>0.2</td>
<td>2</td>
<td>0.2</td>
<td>2</td>
<td>0.1</td>
<td>2</td>
<td>0.1</td>
<td>2</td>
</tr>
<tr>
<td>0.1</td>
<td>1</td>
<td>0.2</td>
<td>1</td>
<td>0.1</td>
<td>1</td>
<td>0.1</td>
<td>1</td>
</tr>
<tr>
<td>0.0</td>
<td>0</td>
<td>0.0</td>
<td>0</td>
<td>0.0</td>
<td>0</td>
<td>0.0</td>
<td>0</td>
</tr>
</tbody>
</table>
### TABLE 2

**Summary of Calculations**

<table>
<thead>
<tr>
<th>Amount $C$ of resources available for the remaining stages of the allocation process</th>
<th>Number of stages remaining in the allocation process</th>
<th>1 Pump</th>
<th>2 Condenser</th>
<th>3 Turbine</th>
<th>4 Reactor core</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$f_1(C)$</td>
<td>$C_P$</td>
<td>$f_2(C)$</td>
<td>$C_C$</td>
<td>$f_3(C)$</td>
</tr>
<tr>
<td>10</td>
<td>80</td>
<td>9</td>
<td>14</td>
<td>3</td>
<td>2.5</td>
</tr>
<tr>
<td>9</td>
<td>80</td>
<td>9</td>
<td>12</td>
<td>3</td>
<td>2.4</td>
</tr>
<tr>
<td>8</td>
<td>70</td>
<td>7 or 8</td>
<td>12</td>
<td>2</td>
<td>2.0</td>
</tr>
<tr>
<td>7</td>
<td>70</td>
<td>7</td>
<td>10</td>
<td>3</td>
<td>1.6</td>
</tr>
<tr>
<td>6</td>
<td>60</td>
<td>6</td>
<td>8</td>
<td>3</td>
<td>1.0</td>
</tr>
<tr>
<td>5</td>
<td>50</td>
<td>4 or 5</td>
<td>5</td>
<td>1</td>
<td>0.8</td>
</tr>
<tr>
<td>4</td>
<td>50</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>0.2</td>
</tr>
<tr>
<td>3</td>
<td>40</td>
<td>3</td>
<td>1</td>
<td>1 or 2</td>
<td>0.2</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0.0</td>
</tr>
<tr>
<td>1</td>
<td>10</td>
<td>1</td>
<td>0</td>
<td>—</td>
<td>0.0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>—</td>
<td>0</td>
<td>—</td>
<td>0.0</td>
</tr>
</tbody>
</table>
Using the formulations in Section 13, develop the return in Table 2 and the optimal spending policy in Table 3.

**TABLE 3**

**Optimal Spending Policies**

<table>
<thead>
<tr>
<th>Total funds available, C</th>
<th>Core $C_R$</th>
<th>Turbine $C_T$</th>
<th>Condenser $C_C$</th>
<th>Pump $C_P$</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>9</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>9</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3, 2, 1, 0</td>
<td>Insufficient</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Answer the following questions.

(a) What is the maximum system efficiency obtainable by spending an amount of money $C$?

(b) How are the funds $C$ to be allocated among the four components in order to produce maximum system efficiency?

(c) Is it ever possible to obtain the same maximized system efficiency for less than the total amount of money $C$ allowed to be spent?

(d) How sensitive is the maximized system efficiency to variations in money spent?

(e) Is there a point of diminishing return beyond which an increase in money spent will not result in an increased system efficiency?

(Kallay, N. [9].)

Define

\[ G = \text{primary coolant flow with } N \text{ stages ahead} \]
\[ N = \text{number of stages} \]
\[ x = \text{flow branching off at first vertical pipe} \]
\[ (G - x) = \text{flow with } (N - 1) \text{ stages ahead} \]
\[ \phi_i(G, x) = \text{work done by primary coolant in the } i\text{th stage with} \]
\[ \text{entering flow } G \text{ and branch flow } x \]
\[ f_N(G) = \text{minimum work done by total flow } G \text{ over } N \text{ stages} \]

Under suitable assumptions, it can be shown that

\[ \phi_i(G, x) = HG^3 + Vx^3 \]

where \( H \) and \( V \) are constants for the horizontal and vertical sections of the exchanger.

Show that the flow distribution through the net work for minimum expenditure of work against friction is given by

\[ f_N(G) = \min_{0 \leq x \leq G} [\phi_N(G, x) + f_{N-1}(G - x)] \]
\[ f_1(G) = \phi_1(G) \]

(Kallay, N. [9].)

25. A plant operator has an initial sum of money \( x \) to be invested in a process that suffers from fast technological changes. He can divide his investment between two types of plants which earn \( g(y) \) for one plant and \( h(x - y) \) for the other. After one time period, the first plant has a salvage value of \( ay \) and the second plant has a salvage value of \( b(x - y) \). Show that the functional equation for the return over \( N \) stages is

\[ f_N(x) = \max_{0 \leq y \leq x} [g(y) + h(x - y) + f_{N-1}(ay + b(x - y))] \]
\[ f_1(x) = \max_{0 \leq y \leq x} [g(y) + h(x - y)] \]

\[ 0 \leq a, b \leq 1 \]

If the operator is not required to spend all of his funds but may hold a portion in reserve, show that

\[ f_N(x) = \max [g(y_1) + h(y_2) + f_{N-1}(ay_1 + by_2 + c(x - y_1 - y_2))] \]

where

\[ y_1 + y_2 \leq x; \quad y_1, y_2 \geq 0; \quad c = \text{interest rate} \]
3. ALLOCATION PROBLEMS

Suppose that the operator receives supplementary funds \( r_k, k = 1, 2, \ldots \), at the beginning of the \( k \)-th period to augment his initial budget \( x \), show that, if he is not required to spend all his funds,

\[
f_N(x) = \operatorname{Max} [g(y_1) + h(y_2) + f_{N-1}(ay_1 + by_2 + c(x - y_1 - y_2) + r_{N-1})]
\]

where

\[
y_1 + y_2 \leq x; \quad y_1, y_2 \geq 0
\]

Taking into account the present worth of money \( \alpha \), show that the

\[
f_N(x) = \operatorname{Max} [g(y_1) + h(y_2) + \alpha f_{N-1}(ay_1 + by_2 + c(x - y_1 - y_2) + r_{N-1})]
\]

(Bellman, R., On some application of the theory of dynamic programming to logistics. The RAND Corporation, P-457 (November 19, 1953).)
1. Introduction

A great many interesting and challenging problems fall under the aegis of the calculus of variations. These problems are characterized by extremizing a quantity involving integrals. Typical industrial problems are production control, inventory control, guidance problems, minimization of costs or maximization of profits, the allocation of resources, and the "best" path problems.

In this chapter, we shall deal with the calculus of variations and its relationship with dynamic programming. At various places, we will solve the same problem by both techniques to show the two different points of view and how they can be reconciled.

In addition to solving the same problem by both the calculus of variations and dynamic programming, we will discuss how to solve the same problem by both a continuous dynamic programming method and a discrete dynamic programming method.

Various difficulties associated with the calculus of variations approach will be discussed. In the course of this discussion, the need for other methods will become apparent. The role of dynamic programming in meeting the deficiencies in the classical calculus of variations is pointed out.

To review the chapter briefly, we discuss in Section 2 a typical one-dimensional calculus of variations problem. Familiar terms and condi-
tions in the calculus of variations such as the Euler-Lagrange equations, a variation, and two-point boundary values are identified. A comparison of the differential calculus and the calculus of variations optima is made in Section 3. Sections 4–11 describe certain difficulties in the calculus of variations. Included among these are constraints, two-point boundary-value problems, linearity, and nonanalytical functions. Various devices to deal with inequality constraints are presented in Section 7. An example is given in Section 8 of the difficulties associated with two-point boundary-value problems. The inconsistency in the Euler-Lagrange expression for linear functions is demonstrated in Section 9. Since the calculus of variations deals with analytical functions, we might very well expect difficulties, as shown in Section 10, with extremizing a nonanalytical function, such as the absolute value of a function. To emphasize the difficulties of the calculus of variations approach, the catalyst replacement problem discussed earlier in the text is reformulated as a calculus of variations problem in Section 11. The difficulty in solving the Euler-Lagrange equations that arise is discussed. In Section 12, we give a brief exposition of the dynamic programming point of view to variational problems. Following this, in Sections 13–16 we derive the continuous dynamic programming equations for several types of variational problems. Having developed the maximum of

\[ J(y) = \int_0^\infty F(x, y) \, dt \]

by dynamic programming in Section 15, we optimize \( J(y) \) in Section 17 by the calculus of variations. The reconciliation of the calculus of variations global point of view and the dynamic programming local point of view is accomplished in Section 18 by the derivation of the Euler-Lagrange equations from the dynamic programming equations. An equation which often appears as the result of treating variational problems by dynamic programming is

\[ f_T = \max_v [F(c, v) + G(c, v) f_c] \]

Its numerical evaluation is treated in Section 19. Section 20 describes the dynamic programming finite analog to the maximization of \( \int_0^T F(x, y) \, dt \), which was previously handled in Section 14 by continuous dynamic programming. In Section 21, a finite analog to the minimization of \( \int_0^T F(y, y', t) \, dt \) is detailed. The application of the methods developed in the chapter to the control of a chemical reactor may be found in Section 22. The method of characteristics is used to solve the reactor
2. PROBLEMS

2. Calculus of Variations Problems

The calculus of variations deals with problems in which an integral involving a function and its derivatives must be extremized. A typical formulation is that of minimizing the quantity

\[ J = \int_{0}^{T} F(y, y', t) \, dt \]  

(1)

where the function \( F(y, y', t) \) can be considered in the process industries applications as a cost function. The function \( y \) and its derivative with respect to time \( y' \) may be thought of as a manipulated variable such as feed rate, temperature, or pressure.

The solution to this problem, if it exists, is customarily found from the solution of the celebrated Euler-Lagrange equation, associated with the expression in Eq. (1). This equation is developed by using a sequence of approximating curves to the true minimizing curve. In many cases, the solution of the Euler-Lagrange equation yields the variables \( y \) and \( y' \) as functions of \( t \) that minimize Eq. (1). Details of the development of the Euler-Lagrange equation can be found in any standard calculus of variations text [28–30, 33, 40, 52].

In one variable, the limiting process described above yields the equation:

\[ \frac{d}{dt} \frac{\partial F}{\partial y'} - \frac{\partial F}{\partial y} \frac{\partial y}{\partial y'} \bigg|_{t=0} = 0 \]  

(2)

Since the boundary conditions usually are chosen so that the third term is equal to zero, we may write:

\[ \frac{d}{dt} \frac{\partial F}{\partial y'} - \frac{\partial F}{\partial y} = 0 \]  

(3)

This is the Euler-Lagrange equation.
The boundary condition term in (2), namely, \( \frac{\partial F}{\partial y'} \delta y \bigg|_{t=0}^T \), equals zero due to either the term \( \frac{\partial F}{\partial y'} = 0 \) or the variation \( \delta y = 0 \) at the end points. There are four possible end conditions to cause the boundary condition term to go to zero.

A variation, expressed here as \( \delta y \), represents the difference between adjacent curves approximating the minimizing curve. There is a difference between a variation and a derivative. A derivative represents an infinitesimal change along a curve, while a variation represents an infinitesimal difference between adjacent curves. For example, if the approximating curve and the minimizing curve pass through the same point, perhaps an end point, the difference between the two curves at this point is zero and the variation \( \delta y = 0 \).

There are as many Euler-Lagrange equations as there are dependent variables. If, for example, the function to be minimized is

\[
J = \int_0^T F(y_1, y_2, y_1', y_2', t) \, dt
\]

there are two Euler-Lagrange equations

\[
\frac{d}{dt} \frac{\partial F}{\partial y_1'} - \frac{\partial F}{\partial y_1} = 0
\]

(5)

\[
\frac{d}{dt} \frac{\partial F}{\partial y_2'} - \frac{\partial F}{\partial y_2} = 0
\]

(6)

The Euler-Lagrange equations are, in general, in the foregoing case a system of second-order nonlinear differential equations. In most instances, they cannot be solved analytically. The equations involving two-point boundary values require for numerical solution a trial and error method in order to satisfy the boundary conditions. Typical boundary conditions for \( y_1 \) and \( y_2 \) are

\[
t = 0 \quad \delta y_1 = 0, \quad t = T \quad \delta y_1 = 0
\]

(7)

\[
t = 0 \quad \delta y_2 = 0, \quad t = T \quad \frac{\partial F}{\partial y_2'} = 0
\]

(8)

Frequently the physical nature of the problem indicates which of the four possible boundary conditions apply to \( y_1 \) and \( y_2 \). If the function depends on many variables, then a rather difficult problem of solving a large set of nonlinear differential equations with two-point boundary values must be faced.
3. Ordinary Maxima and Minima and the Calculus of Variations

It is important to recognize the difference between obtaining the instantaneous minimum of $F(y, y', t)$ and the minimum of $\int_0^T F(y, y', t) \, dt$. The instantaneous minimum can be found by the ordinary differential calculus from the necessary condition:

$$\left( \frac{\partial F}{\partial y} \right) y' + \left( \frac{\partial F}{\partial y'} \right) y'' + \frac{\partial F}{\partial t} = 0 \tag{1}$$

The solution to this equation will not, in general, be the same as the solution to Euler-Lagrange equation. The difference is due, of course, to minimizing instantaneously, where the immediate return is the only goal of importance, in contrast to minimizing over time. In the course of minimizing over time, we may incur higher costs now to ensure lower costs in the future. That is to say, instantaneous goals, in general, are not compatible with over-all goals.

In chemical reactors, for example, we may punish the catalyst by operating at extreme conditions to obtain high productivity for the moment. In the long run, the over-all cost may be increased due to the effects of the "instantaneous" optimization of the process.

4. Difficulties in the Calculus of Variations

The solutions of calculus of variations problems often are difficult to obtain. These troubles range from difficulties in setting up the problems to not being able to solve the Euler-Lagrange equations once they are available.

We will discuss first difficulties which frustrate or at least immobilize the classical calculus of variations. These difficulties are: constraints, two-point boundary value problems, linearity, and unusual or non-analytical functions. Later on, we will discuss what can be done about these difficulties using dynamic programming.

Unfortunately, none of the standard texts on calculus of variations mention these difficulties, or to be generous, pay more than passing attention to them. The resolution of the difficulties cited above is especially important in practical applications where inequality constraints may be the rule rather than the exception. In addition, solutions in principle are not as good as solutions in fact, especially where numerical results must be obtained. The importance of generating computable formulations cannot be overemphasized.
5. Equality Constraints

The expression \( \int_T^0 F(y_1, y_2, y_1', y_2', t) \, dt \) may be minimized subject to equality constraints or relationships such as

\[
y_1 + y_2^2 = C_1
\]  

(1)

This may be handled in the calculus of variations satisfactorily by using a Lagrangian multiplier \( \lambda(t) \) and forming the new function

\[
L = F(y_1, y_2, y_1', y_2', t) + \lambda(t)(y_1 + y_2^2 - C_1)
\]

where \( \lambda(t) \) is a function of time to be determined by the solution of the problem. The Euler-Lagrange expressions become

\[
\frac{d}{dt} \frac{\partial L}{\partial y_1'} - \frac{\partial L}{\partial y_1} = 0
\]  

(2)

\[
\frac{d}{dt} \frac{\partial L}{\partial y_2'} - \frac{\partial L}{\partial y_2} = 0
\]  

(3)

The Euler-Lagrange equations plus Eq. (1) and the boundary conditions are sufficient under appropriate conditions to generate the optimizing expressions for \( y_1, y_2, \) and \( \lambda(t) \).

Equality constraints may turn up due to material or heat balance considerations. If it is not possible to substitute these relations directly into the \( F(y_1, y_2, y_1', y_2', t) \) expression, then the Lagrangian multiplier must be introduced. Where at all convenient, the equality constraints should be substituted directly into the \( F \) expression to reduce the number of equations and number of variables.

If the expression \( \int_0^T F(y_1, y_2, y_1', y_2', t) \, dt \) is to be minimized subject to the equality isoperimetric constraint

\[
\int_0^T G(y_1, y_2, y_1', y_2', t) \, dt = C_2
\]  

(4)

a new function, \( H \), is created:

\[
H = F(y_1, y_2, y_1', y_2', t) + \lambda G(y_1, y_2, y_1', y_2', t)
\]  

(5)

Here the Lagrangian multiplier \( \lambda \) is not a function of time, but it is a constant which must be found as a consequence of the solution of the
Euler-Lagrange equations. The solution is found from the equations

\[
\frac{d}{dt} \frac{\partial H}{\partial y'_1} - \frac{\partial H}{\partial y_1} = 0
\]

and from Eq. (4).

The equality isoperimetric constraint occurs very naturally in practical problems where the total resource or total energy over the entire time period must be equal to a fixed quantity. The isoperimetric constraint is a condition which must be satisfied over the entire interval of time, while the equality constraint must be satisfied at each instant of time. In many applications, the equality constraint and the equality isoperimetric constraint can be handled quite satisfactorily by the formal methods of the calculus of variations.

6. Inequality Constraints

In many industrial and practical applications, inequality constraints of the type

\[(y_1)_{\text{Min}} \leq y_1 \leq (y_1)_{\text{Max}}\]  

or of the type

\[y_1 + y_2 \leq C\]

occur very naturally. The \(y_1\) or \(y_2\) might represent temperatures, pressures, or flows that are constrained due to equipment limitations, chemical purity limitations, economic limitations, or explosive hazards. Since the constraints are truly meaningful, the mathematical optimization procedure should recognize their existence and honor their presence in the solution.

It is also possible to have an inequality isoperimetric constraint of the form

\[\int_0^T G(y_1, y_2, y'_1, y'_2, t) \leq C_1\]

This type of constraint appears, say, when the utilization of the resource, represented by \(G\), must not exceed the total available resource, \(C_1\).

The handling of inequality constraints within the framework of the formal calculus of variations is generally very difficult for reasons to be discussed next.
The principal difficulty in handling inequality constraints is that the solution consists of a function constrained for part of the time and unconstrained for the rest of the time. The Euler-Lagrange equation describes the $y$ as a function of time only for the free variation. To illustrate the matter, let us consider Fig. 1, constraint (1), and the function $\int_0^TF(y, y', t)\,dt$ to be minimized. Let us consider further that we know that the initial value of $y = y_{\text{Max}}$ and that the final value of $y = y_{\text{Min}}$. We do not know the point $t = t_1$ where the $y$ breaks away from its maximum and we do not know the point $t = t_2$ where $y$ reaches the lower limit. The Euler-Lagrange equation tells us only the shape of the free variation between these limits. In other words, the Euler-Lagrange equation yields a solution as if $t_1 = 0$ and $t_2 = T$. It is not difficult to imagine solutions that look like those in Fig. 2. A problem with multiple breakpoints and multiple free variations represents quite a challenge to classical variational theory. If we consider multi-dimensional problems where $y_1, y_2, ..., y_k$ variables are constrained, the enormity of the difficulties is staggering.

In the last two examples and figures, we have presumed a knowledge of the solution in order to draw the curves. When we first set up the problem, we have little or no idea of the shape of the solution, the number of breakpoints, and the number of free variations. This information is the result of the solution. It may very well happen that for a certain choice of physical parameters the constraints normally present may not come into play. For example, in Fig. 1, $t_1 = 0$, and $t_2 = T$. In other cases, one constraint may so dominant that

\begin{align*}
y_1 &= y_{\text{Max}}, \quad t = 0 \quad \text{to} \quad t = T - \Delta \\
y_1 &= y_{\text{Min}}, \quad t = T - \Delta \quad \text{to} \quad t = T
\end{align*}

where $\Delta$ represents a small element of time.
To summarize, the crux of the difficulty with the inequality constrained problem is that we do not know beforehand (before solving the problem) when the constraints will come into play (if at all). The Euler-Lagrange equations in general do not provide information about the breakpoints.†

![Diagram of inequality constraints]

**FIG. 2.**

### 7. Devices for Handling Inequality Constraints

**a. Bliss**

The simplest dodge for handling an inequality constrained problem, or any constrained problem for that matter, is to set up the Euler-Lagrange equations without considering the constraints. The examination of the results will determine what constraints, if any, were violated. This approach, while blissful, at times does yield interesting results and does offer a basis of comparison between the constrained and unconstrained versions of the problem.

**b. Positivity**

A suitable substitution for constraints of the type

\[ y_{\text{Min}} \leq y \leq y_{\text{Max}} \]  \hspace{1cm} (1)

guarantees that the constraints will be honored. If we let

\[ u^2 = (y - y_{\text{Min}})(y_{\text{Max}} - y) \]  \hspace{1cm} (2)

† In all fairness to the calculus of variations, we should mention that the Erdman corner conditions do handle problems where there is a break or discontinuity in the curve, provided certain boundary conditions are given.
we know that $u^2$ will be equal to zero or positive. When
\[ y = y_{\text{Min}} \quad \text{or} \quad y_{\text{Max}}, \quad u^2 = 0 \] (3)

In the region between $y_{\text{Min}}$, and $y_{\text{Max}}$, $u^2$ is positive. This device is due to Valentine [51].

c. Augmented Objective Function

A numerical device used to guarantee that the inequality constraints are respected is to augment the objective function by an additional term which penalizes the solution so heavily for a constraint violation that it never occurs. Suppose it is desired to minimize
\[ J = \int_0^T F(y, y', t) \, dt \] (4)
subject to
\[ 0 \leq y < c \] (5)

We augment the $F(y, y', t)$ by adding the term $\lambda(y/c)^m$, so the new objective to be minimized is
\[ J = \int_0^T \left[ F(y, y', t) + \lambda \left( \frac{y}{c} \right)^m \right] dt \] (6)

Here $\lambda$ and $m$ are arbitrarily chosen large numbers. As long as $y$ is small compared to $c$, the fraction $y/c$ raised to the $m$ power is very small and the product $\lambda(y/c)^m$ also is small. When $y$ approaches $c$, the ratio $y/c \to 1$ and the product $\lambda(y/c)^m \to \lambda$. Since $\lambda$ is a large number, as $y \to c$ or $y$ exceeds $c$, the term $\lambda(y/c)^m$ becomes very large and dominates $F(y, y', t)$.

Now if we consider $\lambda$ to be the cost per unit of $y$, the term $\lambda(y/c)^m$ represents the cost of utilizing $y$. Since the objective is to minimize $\int_0^T F(y, y', t) \, dt$ and since the augmented objective is to minimize
\[ \int_0^T \left[ F(y, y', t) + \lambda \left( \frac{y}{c} \right)^m \right] dt \]
the augmented objective approaches the original objective provided $\lambda(y/c)^m$ is small.
This device provides a suitable technique for solving inequality constrained problem by the classical calculus of variations. The choice of the numerical values for $\lambda$ and $m$ is found from computational experience.

d. Special Function

Another device, used successfully by Miele, is to define a function $\beta$ in the interval $(0, 1)$ as follows:

\begin{align*}
\beta(y) &= 0, \quad y < y_{\text{Min}} \\
0 &\leq \beta(y) \leq 1, \quad y_{\text{Min}} \leq y \leq y_{\text{Max}} \\
\beta(y) &= 1, \quad y > y_{\text{Max}}
\end{align*}

The function $\beta$ has the property of being positive with values between 0 and 1 over all values of $y$ (see Fig. 3). It also guarantees that the constraints will be honored when $\beta$ is substituted for $y$. In Miele's trajectory problems, he had enough physical insight into the problem to establish the breakpoints [44-46].

e. Neyman-Pearson Lemma

Certain problems in the calculus of variations have been solved by employing the Neyman-Pearson lemma. These results are not easily obtained and as yet are not within the everyday use of engineering. References [5, 7, 20] at the end of the chapter may be consulted.
8. Two-Point Boundary Value Problems

In general, the Euler-Lagrange equations are second degree or higher nonlinear differential equations with two boundary values to be satisfied. The boundary conditions in one variable are given by

\[ t = 0, \quad \delta y = 0 \quad \text{or} \quad \frac{\partial F}{\partial y'} = 0 \]

\[ t = T, \quad \delta y = 0 \quad \text{or} \quad \frac{\partial F}{\partial y'} = 0 \]  

Any two of the four possible boundary conditions at \( t = 0 \) and \( t = T \) are possible; for example,

\[ t = 0, \quad \delta y = 0 \]  

\[ t = T, \quad \frac{\partial F}{\partial y'} = 0 \]

would be one possible set of boundary conditions. In some problems enough is known about the problems to specify the boundary conditions prior to the solution. In other problems, an examination of the implications of the boundary conditions helps to determine what boundary conditions should prevail.

As long as it is possible to generate analytical solutions to the Euler-Lagrange equations, the two-point boundary value aspects offer no difficulty. The real difficulty arises when analytical results cannot be obtained and numerical methods must be resorted to.

The problem may be stated as follows: "Solve the Euler-Lagrange differential equations subject to the given boundary conditions at \( t = 0 \), and at \( t = T \)." We know from the boundary conditions only two points on the extremal curve. What we do not know is how to move from either known point to the other along the extremal. The information that is lacking is the derivative at \( t = 0 \) or \( t = T \). In the case of one dependent variable, to initiate a numerical solution we must assume a value of the derivative at, say, \( t = 0 \) and from it generate a curve of \( y \) versus \( t \) so that the boundary condition at \( t = T \) is satisfied. To illustrate, suppose the Euler-Lagrange equation is

\[ \frac{d^2 y}{dt^2} = y^2 + t^2, \]
and the boundary conditions are given

\[
\begin{align*}
t & = 0, \quad y = y_0 \\
t & = T, \quad y = y_F
\end{align*}
\]

Let us first subdivide the time interval \((0, T)\) into \(N\) elements of length \(\Delta\), so that \(N\Delta = T\). The term \(y(k\Delta)\) refers to the value of \(y\) at the \(k\)th time interval measured from \(t = 0\).

We let, for convenience,

\[
y_k = y(k\Delta), \quad k = 0, 1, 2, ..., N
\]

The first derivative is approximated by

\[
\frac{dy}{dt} \approx \frac{y_{k+1} - y_k}{\Delta}, \quad k = 0, 1, 2, ..., N
\]

The second derivative is approximated by

\[
\frac{d^2y}{dt^2} \approx \frac{y_{k+1} - y_k - y_k - y_{k-1}}{\Delta} = \frac{y_k - y_{k-1}}{\Delta}
\]

Equation (4) may be approximated by

\[
\frac{y_{k+1} - y_k - y_k - y_{k-1}}{\Delta} = \frac{y_k^2 + \Delta^2}{\Delta}
\]

If we assume that at \(k = 0\)

\[
\frac{y_0 - y_{-1}}{\Delta} = A_0 = \text{constant} = \text{initial derivative estimate}
\]

then we may generate the solution to the approximating differential equation

\[
\frac{y_1 - y_0 - A_0}{\Delta} = \frac{y_0^2 + \Delta^2}{\Delta}
\]

Since \(y_0\), \(\Delta\), and \(A_0\) are known, \(y_1\) may be found.

For the next interval, Eq. (4) is approximated by

\[
\frac{y_2 - y_1 - y_1 - y_0}{\Delta} = \frac{y_1^2 + \Delta^2}{\Delta}
\]
Since \( y_0, \Delta, \) and \( y_1 \) are known, \( y_2 \) may be found. This process is continued step by step until the time \( (k = N) \). If the initial derivative at \( k = 0 \) is chosen correctly, then \( y_N \) equals \( y_F \). If the initial derivative is not chosen correctly, \( y_N \neq y_F \). A new estimate of the initial derivative must be made and the process repeated until satisfactory closure is obtained.

The objections to this trial and error process are inherent in any numerical solution to boundary value problems whether associated with the calculus of variations or not. The first problem is that of choosing a suitable approximating equation. The example given above utilizes only one of many possibilities. The second is the choosing of a suitable increment, \( \Delta \). The first and second problems are closely related since numerical solutions to differential equations are plagued by the specter of convergence and stability. The third problem is how to choose the next initial derivative approximation value in order to reduce the number of trials. The fourth problem concerns multi-dimensional problems where the example given above must be worked out in multiple variables. In these problems it may very well happen that for some choices of initial derivative approximations that the terminal boundary conditions are met for some variables but not for all. To meet all the terminal boundary conditions may require as much patience as computing time. The fifth problem is that once the boundary conditions are satisfied, one is not really sure of the uniqueness of the answer.

9. Linearity

There are some problems for which Euler-Lagrange equations give no useful information. For example, let us consider minimizing \( \int_0^T F(y, y') \, dt \), where \( F \) is defined by

\[
F = y + y' \tag{1}
\]

Then

\[
\frac{\partial F}{\partial y'} = 1, \quad \frac{\partial F}{\partial y} = 1 \tag{2}
\]

and the Euler-Lagrange equation yields an expression that is inconsistent:

\[
\frac{d}{dt} (1) - 1 = 0 \tag{3}
\]

\[
0 - 1 = 0 \tag{4}
\]
10. Nonanalytic Functions

In some problems, particularly in the control field, we are called upon to extremize unusual functions such as

\[ J = \int_0^T |x| \, dt \]

subject to certain constraints, where \( |x| \) means the absolute value of \( x \). The classical calculus of variations is not able to handle problems of this type since it requires among other conditions continuous derivatives.

Since dynamic programming often employs numerical methods for solving these optimization problems, the requirement of minimizing the absolute value of \( x \) is numerically really only slightly different from minimizing positive \( x \) or minimizing negative \( x \).

11. Example: Inequality Constrained Problem

To illustrate more fully some of the difficulties encountered with inequality constraints in the calculus of variations, we will consider a version of the catalyst replacement problem. Using the positivity device of Section 7b to eliminate the inequality constraints, we set up the Euler-Lagrange equations.

Referring to the system described in Chapter 2, Sections 5-7, we desire to maximize the profit of operating the chemical reactor over the interval \((0, t_1)\), namely, \( \int_0^{t_1} P \, dt \)

The integral is subject to the relationships

\[ C = c_1T^M - c_2F - c_3 \int_0^t F \, dt \]  \( \text{(1)} \)

where \( M \) is a known constant

\[ Q = Fc_p(T - T_0) + HCF \]  \( \text{(2)} \)
The following constraints must be observed:

\[ C_L \leq C \leq C_U \]  
\[ F_L \leq F \leq F_U \]  
\[ T_L \leq T \leq T_U \]  

The subscripts \( U \) and \( L \) refer to the upper and lower limits.

The profit is described by

\[ P = CFV_1 - (1 - C)FV_2 - FV_3 - QV_4 \]  

The boundary conditions are

\[ t = 0, \quad \int_0^t F \, dt = 0 \]  
\[ t = t_f, \quad \int_0^{t_f} F \, dt = K_1 = \text{constant} \]  

The following constants are known: \( c_1, c_2, c_3, M, c_p, T_0, H, C_L, C_U, F_L, F_U, T_L, T_U, V_1, V_2, V_3, V_4, K_1, \) and \( t_f \).

Substitution of Eqs. (1) and (2) into (6) gives

\[ P = c_1(V_1 - V_2 - HV_4)FTM - c_2(V_1 - V_2 - HV_4)F^2 \]  
\[ - c_3(V_1 - V_2 - HV_4)F \int_0^t F \, dt + F(V_2 - V_3 - c_pV_4T + c_pT_0V_4) \]  
\[ P = a_1FTM - a_2F^2 - a_3F \int_0^t F \, dt + F(a_4 - c_pV_4T) \]  

where

\[ a_1 = c_1(V_1 - V_2 - HV_4) \]  
\[ a_2 = c_2(V_1 - V_2 - HV_4) \]  
\[ a_3 = c_3(V_1 - V_2 - HV_4) \]  
\[ a_4 = V_2 - V_3 + c_pT_0V_4 \]

We now define some new variables

\[ z_C^2 = (C_U - C)(C - C_L) \]  
\[ z_F^2 = (F_U - F)(F - F_L) \]  
\[ z_T^2 = (T_U - T)(T - T_L) \]
and let

$$
\varphi_1 = z^2_C - (C_U - C)(C - C_L)
$$

$$
\varphi_2 = z^2_F - (F_U - F)(F - F_L)
$$

$$
\varphi_3 = z^2_T - (T_U - T)(T - T_L)
$$

A new function $G$ is formed:

$$
G = P + \lambda_1(t)\varphi_1 + \lambda_2(t)\varphi_2 + \lambda_3(t)\varphi_3
$$

Our objective is now to maximize $\int_0^t G \, dt$

On substituting Eq. (10) into Eq. (21), we have

$$
G = a_1F T^M - a_2F^2 - a_3F \int_0^t F \, dt + F(a_4 - c_2V_4T)
$$

$$
+ \lambda_1(t)\varphi_1(z_C, C) + \lambda_2(t)\varphi_2(z_F, F) + \lambda_3(t)\varphi_3(z_T, T)
$$

We let

$$
q = \int_0^t F \, dt
$$

$$
q' = F
$$

Substituting Eqs. (23) and (24) into (22), we find that

$$
G = a_1q'T^M - a_2(q')^2 - a_3q'q + q'(a_4 - c_2V_4T)
$$

$$
+ \lambda_1(t)\varphi_1(z_C, C) + \lambda_2(t)\varphi_2(z_F, q') + \lambda_3(t)\varphi_3(z_T, T)
$$

To simplify the problem, we drop $\varphi_1$.

We now form the Euler-Lagrange equations with respect to $T, q, z_F, z_T$:

$$
\frac{\partial G}{\partial T} = a_1Mq'T^{M-1} - c_2V_4q' + \lambda_3(t)(2T - T_U - T_L) = 0.
$$

Since $t$ does not occur explicitly in Eq. (25), a first integral of the Euler-Lagrange equation with respect to $q$ is

$$
G - q' \frac{\partial G}{\partial q'} = K_2 = \text{constant}
$$
Writing out Eq. (27) in detail as Eq. (28) and simplifying in Eq. (29), we have

\[ a_1 q'T^M - a_3 (q')^2 - a_5 q' + q' (a_4 - c_2 V_4 T') \]
\[ + \lambda_2(t)[z_F^2 - (F_U - q') (q' - F_L)] + \lambda_3(t)[z_T^2 - (T_U - T) (T - T_L)] \]  
(28)

\[- q'[a_1 T^M - 2a_2 q' - a_5 q + (a_4 - c_2 V_4 T') + \lambda_2(t)(2q' - F_U - F_L)] = K_2 \]

\[ a_3 (q')^2 + \lambda_2(t)[z_F^2 - (F_U - q') (q' - F_L)] + \lambda_3(t)[z_T^2 - (T_U - T) (T - T_L)] \]
\[ - \lambda_2(t)q'(2q' - F_U - F_L) = K_2 \]  
(29)

\[ \frac{\partial G}{\partial z_F} = 2\lambda_2(t)z_F = 0 \]  
(30)

\[ \frac{\partial G}{\partial z_T} = 2\lambda_3(t)z_T = 0 \]  
(31)

Summarizing the Euler-Lagrange equations, we have

\[ a_1 Mq'T^M-1 - a_2 V_4 q' + \lambda_3(t)(2T - T_U - T_L) = 0 \]  
(26')

\[ a_3 (q')^2 + \lambda_2(t)[z_F^2 - (F_U - q') (q' - F_L)] + \lambda_3(t)[z_T^2 - (T_U - T) (T - T_L)] \]
\[ - \lambda_2(t)q'(2q' - F_U - F_L) = K_2 \]  
(29)

\[ \lambda_2(t)z_F = 0 \]  
(30')

\[ \lambda_3(t)z_T = 0 \]  
(31')

Between the Euler-Lagrange equations and the \( z_T \) and \( z_F \) equations

\[ z_F^2 - (F_U - F) (F - F_L) = z_F^2 - (F_U - q') (q' - F_L) = 0 \]  
(16')

\[ z_T^2 - (T_U - T) (T - T_L) = 0 \]  
(17')

we have six equations in the six unknowns: \( q', T, z_F, z_T, \lambda_2(t), \) and \( \lambda_3(t). \)

Equations (16'), (17'), (26'), (29), (30'), and (31') can be reduced to four equations in four unknowns by eliminating \( z_F \) and \( z_T: \)

\[ a_1 Mq'T^M-1 - a_2 V_4 q' + \lambda_3(t)(2T - T_U - T_L) = 0 \]  
(26')

\[ a_3 (q')^2 - \lambda_2(t)q'(2q' - F_U - F_L) = K_2 \]  
(29')

\[ \lambda_2(t) \sqrt{(F_U - q') (q' - F_L)} = 0 \]  
(30'')

\[ \lambda_3(t) \sqrt{(T_U - T) (T - T_L)} = 0 \]  
(31'')}
Up to this point, we have set up the Euler-Lagrange equations in the classical manner. Now we face the problem of solving the equations. Inspection of the last four equations shows there is no way to uncouple the Lagrangian multiplier from the square root term in Eqs. (30'') and (31''). Equations (30'') and (31'') state that the product over the permissible time span must be equal to zero. This includes the possibilities of \( \lambda_2(t) = 0 \) for all time, \( \lambda_2(t) = 0 \) for part of the time, \( \lambda_2(t) \neq 0 \) for all of the time. The same is true for \( \lambda_3(t) \). Another difficulty here is the impossibility of advancing \( \lambda_2(t) \) and \( \lambda_3(t) \) as functions of time. Normally this is accomplished through the presence of derivatives for \( \lambda_2(t) \) and \( \lambda_3(t) \) or through derivatives of other variables in the set of equations. Here the absence of derivatives for \( \lambda_2(t) \) and \( \lambda_3(t) \) as well as the relationships among the variables themselves preclude generating \( \lambda_2(t) \) and \( \lambda_3(t) \).

While the solution of the Euler-Lagrange equations ordinarily yields the required function of time, in the problem involving inequality constraints, this does not occur. Without some additional insight into the nature of the problem or without additional information supplied, it is a matter of some difficulty to solve this problem with inequality constraints.

12. Dynamic Programming Point of View

The calculus of variations point of view to variational problems is global. By that we mean that the optimum (but unknown) curve is located by means of a succession of approximating curves that satisfy the boundary conditions. In the limit, the approximating curves become the desired extremal. The dynamic programming point of view to variational problems may be called local. Instead of attempting to find the entire extremal at one stroke, dynamic programming evaluates the optimal derivative at each point along the extremal. Knowing a point on the extremal and the optimal derivative, it is possible to move to the next point, where once again the optimal derivative is found.

These two points of view, one global and the other local, yield the same results. As will be shown subsequently, we may derive from the linear partial differential equation of dynamic programming, the Euler-Lagrange equation. We will show also that the local point of view makes it possible for dynamic programming to attack problems that are difficult to treat by means of the formal calculus of variations.

In the following sections, we develop typical dynamic programming solutions and compare the results to the formal results of the calculus of variations.

In the next few sections, we will develop the solution to a class of variational problems of the form

\[ J(y) = \int_0^T F(x, y, t) \, dt \]  

where \( x \) and \( y \) are related by the equation

\[ \frac{dx}{dt} = G[x(t), y(t)] \]  

The initial condition at \( t = 0 \) is

\[ x(0) = c \]

We desire to maximize or minimize \( J(y) \).

As we shall see soon, the dynamic programming approach to problems of the type given in (1)–(3) leads to a nonlinear partial differential equation. This nonlinear partial differential equation may be solved in several ways: by analytical methods, by the method of characteristics, or by finite difference approximation.

The cases we consider are:

(a) \[ J(y) = \int_0^T F(x, y) \, dt \]

(b) \[ J(y) = \int_0^\infty F(x, y) \, dt \]

(c) \[ J(y) = \int_0^T F(x, y, t) \, dt \]

14. \( \text{Max} \int_0^T F(x, y) \, dt \)

We desire to maximize \( J(y) \) where

\[ J(y) = \int_0^T F(x, y) \, dt \]

over all \( y \), with

\[ \frac{dx}{dt} = G(x, y) \]
14. \( \max \int_0^T F(x, y) \, dt \)

The initial condition is

\[ x(0) = c \]  

We define

\[ f(c, T) = \max_y J(y) = \max_y \int_0^T F(x, y) \, dt \]

The state of system is specified by the initial value of \( x \), namely, \( c \), and the duration of the process, \( T \). The maximization is executed by choosing the proper value of \( y \) as a function of the state. The term \( y[0, T] \) refers to the choice of \( y \) in the interval \((0, T)\).

If we break the time interval into two parts \([0, S]\) and \([S, T]\) we may write

\[ f(c, T) = \max_y \left[ \int_0^S F(x, y) \, dt + \max_y \int_S^T F(x, y) \, dt \right] \]

By the definition of \( f(c, T) \), we note that

\[ f(c(S), T - S) = \max_y \int_S^T F(x, y) \, dt \]

where \( c(S) \) is the value of \( x \) at the time \( S \), and the duration of the process is \((T - S)\). We have by substituting (7) into (6)

\[ f(c, T) = \max_y \left[ \int_0^S F(x, y) \, dt + f(c(S), T - S) \right] \]

By virtue of Eqs. (2) and (3), the term \( c(S) \) may be written

\[ \frac{c(S) - c}{S} = G(x(0), y(0)) = G(c, y(0)) \]

\[ c(S) = c + SG(c, y(0)) \]

to terms of order \( S^2 \).

The significance of the Eqs. (8) and (10) is that the choice of \( y \) in the interval \([0, S]\), transforms \( c \) into \( c(S) \), and the time interval remaining is reduced from \( T \) to \((T - S)\).

Equation (8), using (10), may be cast into the form

\[ f(c, T) = \max_y \left[ \int_0^S F(x, y) \, dt + f(c + SG(c, y(0)), T - S) \right] \]

to terms of order \( S^2 \).
Let us digress for a moment and recall that a function in two variables \( w_0, z_0 \) may be expanded over the interval \( h \) and \( k \) in a Taylor’s expansion as follows:

\[
f(w_0 + h, z_0 + k) = f(w_0, z_0) + h \frac{\partial f}{\partial w_0} + k \frac{\partial f}{\partial z_0} + hk \frac{\partial^2 f}{\partial w_0 \partial z_0} + \ldots \tag{12}
\]

We establish the following correspondence between the terms of \( f(c + SG(c, y(0), T - S)) \) in (11), with the terms in (12):

\[
c \sim w_0 \\
SG(c, y(0)) \sim h \\
T \sim z_0 \\
-S \sim k \\
f(c + SG(c, y(0)), T - S) \sim f(w_0 + h, z_0 + k)
\]

Consider \( S \) to be small, then Taylor’s expansion up to the linear terms of \( f(c(S), T - S) \) is

\[
f(c(S), T - S) = f(c + SG(c, y(0)), T - S) \\
= f(c, T) + \frac{\partial f(c, T)}{\partial c} [SG(c, y(0))] - \frac{\partial f(c, T)}{\partial T} S
\]

Equation (8) is now written for small \( S \):

\[
f(c, T) = \max_{y(0), S} \left[ F(c, y(0)) S + f(c, T) + SG(c, y(0)) \frac{\partial f}{\partial c} - S \frac{\partial f}{\partial T} \right] \tag{15}
\]

Since \( f(c, T) \) is independent of the choice of \( y \), it is brought outside of the brackets on the right-hand side of the equation:

\[
f(c, T) = f(c, T) + \max_{y(0), S} \left[ F(c, y(0)) S + SG(c, y(0)) \frac{\partial f}{\partial c} - S \frac{\partial f}{\partial T} \right] \tag{16}
\]

whence

\[
0 = \max_{y(0), S} \left[ F(c, y(0)) S + SG(c, y(0)) \frac{\partial f}{\partial c} - S \frac{\partial f}{\partial T} \right] \tag{17}
\]

Let

\[
v = y(0) \tag{18}
\]

then in the limit as \( S \to 0 \), Eq. (17) becomes

\[
0 = \max_v \left[ F(c, v) + G(c, v) \frac{\partial f}{\partial c} - \frac{\partial f}{\partial T} \right] \tag{19}
\]
14. \( \max \int_0^T F(x, y) \, dt \)

We observe here that the maximization is now taken over \( v \) since the interval \((0, S)\) has shrunk to 0, and \( v = y(0) \). At the maximum value of \( v \), Eq. (19) leads to two equations:

\[
\frac{\partial f}{\partial T} = F(c, v) + G(c, v) \frac{\partial f}{\partial c} \quad (20)
\]

\[
0 = \frac{\partial F(c, v)}{\partial v} + \frac{\partial G(c, v)}{\partial v} \frac{\partial f}{\partial c} \quad (21)
\]

Equation (21) is found by taking the derivative of Eq. (19) with respect to \( v \). Equation (20) states that once the maximizing value of \( v \) has been found, substitution of it into the right-hand side of Eq. (19) yields the value zero.

Choosing of the proper value of \( v \) is equivalent to choosing the proper value of the derivative \( dx/dt \). Let us note that the derivative in Eq. (2) is expressed for small \( S \) by means of Eq. (10). Since the terms \( c \) and \( S \) are known, the selection of \( y(0) \), that is, \( v \), determines \( G(c, y(0)) \), which is the derivative, evaluated at \( t = 0 \). From this, the next point \( c(S) \) is found, and the procedure is repeated to find the optimal slope at \( c(S) \).

As mentioned earlier, the distinguishing characteristic of the dynamic programming approach to variational problems is the generation of the optimal derivative point by point along the extremal.

The solution of Eq. (19) or its equivalent gives \( v \) as a function of \( c \); that is to say, it gives the initial value of \( y \) at time \( t = 0 \), corresponding to the initial value of \( x \). An alternate way of stating this is that it gives the initial slope of the extremal corresponding to the initial value of \( x \). From Eq. (2), we can find \( x \) and \( y \) as a function of time knowing their dependence upon \( c \) and \( v \).

Equation (19) may be solved by numerical means as discussed in Section 19. It may also be solved by developing and solving the characteristic equations of the equivalent set of equations (20) and (21). This is carried out in the example in Section 22.

A partial differential equation equivalent to Eqs. (20) and (21) may be found by forming and equating \( f_{Te} \) of Eq. (20) and \( f_{cT} \) of Eq. (21). From Eq. (21), we form

\[
\frac{\partial f}{\partial c} = - \frac{\partial F}{\partial v} \frac{\partial v}{\partial G} = P(c, v) \quad (22)
\]

Substituting (22) into (20), we have

\[
\frac{\partial f}{\partial T} = F(c, v) - G(c, v) \frac{\partial F}{\partial v} \frac{\partial v}{\partial G} = Q(c, v) \quad (23)
\]
Using (22) we form
\[ \frac{\partial}{\partial T} \frac{\partial f}{\partial c} = f_T c = \frac{\partial}{\partial T} P(c, v) \] (24)

Since \( v = v(c, T) \) we have
\[ f_T c = \frac{\partial P}{\partial v} \frac{\partial v}{\partial T} \] (25)

Using (23) we form
\[ \frac{\partial}{\partial c} \frac{\partial f}{\partial T} = \frac{\partial}{\partial c} Q(c, v) \] (26)
\[ f_c T = \frac{\partial Q}{\partial c} + \frac{\partial Q}{\partial v} \frac{\partial v}{\partial c} \] (27)

Equating (25) and (27), we have
\[ \frac{\partial P}{\partial v} \frac{\partial v}{\partial T} = \frac{\partial Q}{\partial c} + \frac{\partial Q}{\partial v} \frac{\partial v}{\partial c} \] (28)

The solution of this nonlinear partial differential equation yields the solution of (19). Equation (28) may be solved by the method of characteristics.

15. Max \( \int_0^\infty F(x, y) \, dt \)

We desire to maximize \( J(y) \) where
\[ J(y) = \int_0^\infty F(x, y) \, dt \] (1)
subject to
\[ \frac{dx}{dt} = G(x, y) \] (2)
with the initial condition
\[ x(0) = c \] (3)

This problem differs from the problem in Section 14 only by the upper limit in (1) being equal to infinity.

We define
\[ f(c) = \max_y J(y) = \max_y \int_0^\infty F(x, y) \, dt \]
15. \[ \max \int_0^T F(x, y) \, dt \]

Here the state of the system is specified only by \( c \), the initial value of \( x \). There is no necessity to use two state variables as we did in Section 14, since any infinite time interval is not reduced in duration by an incremental time element \( S \).

In a manner similar to Section 14, we break the time interval into two parts, \([0, S]\) and \([S, \infty)\):

\[
f(c) = \max_{v[0,S]} \max_{v[S,\infty]} \left[ \int_0^S F(x, y) \, dt + \int_S^\infty F(x, y) \, dt \right]
\]

(4)

\[
f(c) = \max_{v[0,S]} \left[ \int_0^S F(x, y) \, dt + \max_{v[S,\infty]} \int_S^\infty F(x, y) \, dt \right]
\]

(5)

By definition of \( f(c) \)

\[
f(c(S)) = \max_{v[S,\infty]} \int_S^\infty F(x, y) \, dt
\]

(6)

Hence, Eq. (5) becomes

\[
f(c) = \max_{v[0,S]} \left[ \int_0^S F(x, y) \, dt + f(c(S)) \right]
\]

(7)

As before, we can expand \( f(c(S)) \) in a Taylor’s series for a small increment \( S \):

\[
f(c(S)) = f(c + G[c, y(0)] \, S) = f(c) + \frac{\partial f}{\partial c} G[c, y(0)] \, S
\]

(8)

Letting \( S \to 0 \), we find that on substituting Eq. (8) into Eq. (7), we have

\[
0 = \max_v \left[ F(c, v) + G(c, v) \frac{\partial f}{\partial c} \right]
\]

(9)

where

\[
v = y(0)
\]

(10)

At the maximum, this equation is equivalent to the set of two equations

\[
0 = F_v(c, v) + G_v(c, v) f'(c)
\]

(11)

\[
0 = F(c, v) + G(c, v) f'(c)
\]

(12)

A new linear partial differential equation is found by eliminating \( f'(c) \) from Eqs. (11) and (12):

\[
F(c, v) G_v(c, v) - G(c, v) F_v(c, v) = 0
\]

(13)

This equation determines \( c \) as a function of \( v \), or \( x \) as a function of \( y \), as explained before. The functions \( x \) and \( y \) can be found as functions of time by means of Eq. (2).
16. Max $\int_0^T F(x, y, t) \, dt$

If we consider $T$ to be fixed, we may imbed the function $\int_0^T F(x, y, t) dt$ into a larger class of problems as follows. Let

$$J(y) = \int_a^T F(x, y, t) \, dt$$

where $a$ ranges from 0, $T$. The constraints are

$$\frac{dx}{dt} = G(x, y, t)$$

$$x(a) = c$$

For fixed $T$, the state variables are $c$, the initial value of $x$, and $a$, the initial time. Let

$$f(c, a) = \max_y J(y) = \max_y \int_a^T F(x, y, t) \, dt$$

As before, we may break the interval of integration into two parts $[a, a+S]$ and $[a+S, T]$: 

$$f(c, a) = \max_{y[a,a+S]} \max_{y[a+S,T]} \left[ \int_a^{a+S} F(x, y, t) \, dt + \int_{a+S}^T F(x, y, t) \, dt \right]$$

$$f(c, a) = \max_{y[a,a+S]} \left[ \int_a^{a+S} F(x, y, t) \, dt + \max_{y[a+S,T]} \int_{a+S}^T F(x, y, t) \, dt \right]$$

From the definition of $f(c, a)$, we observe that

$$f(c(S), a + S) = \max_{y[a+S,T]} \int_{a+S}^T F(x, y, t) \, dt$$

where $c(S)$ is the value of $x$ at time $(a + S)$:

$$f(c, a) = \max_{y[a,a+S]} \left[ \int_a^{a+S} F(x, y, t) \, dt + f(c(S), a + S) \right]$$

For small $S$ we expand

$$f(c(S), a + S) = f(c + SG(c, y(a), a), a + S)$$

$$f(c(S), a + S) = f(c, a) + \frac{\partial f}{\partial c} SG(c, y(a), a) + \frac{\partial f}{\partial a} S$$
16. \( \max \int_0^T F(x, y, t) \, dt \)

Substituting (10) into (8), we have for small \( S \)

\[
f(c, a) = \max_{y \in [a, a+S]} \left[ F(c, y(a), a) S + f(c, a) + SG(c, y(a), a) \frac{\partial f}{\partial c} + S \frac{\partial f}{\partial a} \right]
\]

Set

\[
v = y(a)
\]

Equation (11) becomes, in the limit as \( S \to 0 \),

\[
0 = \max_{v} \left[ F(c, v, a) + G(c, v, a) \frac{\partial f}{\partial c} + \frac{\partial f}{\partial a} \right]
\]

As before, the last equation may be written as two equivalent equations:

\[
0 = \frac{\partial F}{\partial v} + \frac{\partial G}{\partial v} \frac{\partial f}{\partial c}
\]

\[
0 = F + G \frac{\partial f}{\partial c} + \frac{\partial f}{\partial a}
\]

The last two equations may be combined to yield a nonlinear partial differential equation.

From (14) we have

\[
\frac{\partial f}{\partial c} = - \frac{\partial F}{\partial v} \frac{\partial v}{\partial c} = P(c, v, a)
\]

Substitution of (16) into (15) yields

\[
- \frac{\partial f}{\partial a} = F - \frac{G(\partial f/\partial v)}{\partial G/\partial v} = - Q(c, v, a)
\]

Forming \( \partial/\partial a \) (\( \partial f/\partial c \)) from (16), we have

\[
\frac{\partial}{\partial a} P(c, v, a) = \frac{\partial P}{\partial c} \frac{\partial c}{\partial a} + \frac{\partial P}{\partial v} \frac{\partial v}{\partial a} + \frac{\partial P}{\partial a}
\]

Since

\[
v = v(c, a)
\]

we have

\[
dv = \frac{\partial v}{\partial c} dc + \frac{\partial v}{\partial a} da
\]

and

\[
\frac{\partial}{\partial a} P(c, v, a) = \frac{\partial P}{\partial v} \frac{\partial v}{\partial a} + \frac{\partial P}{\partial a} = P_v v_a + P_a
\]
Forming $\partial/\partial c (\partial f / \partial a)$ from (17), we have

$$\frac{\partial}{\partial c} Q(c, v, a) = \frac{\partial Q}{\partial c} + \frac{\partial Q}{\partial v} \frac{\partial v}{\partial c} + \frac{\partial Q}{\partial a} \frac{\partial a}{\partial c}$$  \hspace{1cm} (22)

$$\frac{\partial}{\partial c} Q(c, v, a) = Q_v + Q_0 v_c$$  \hspace{1cm} (23)

Equating (21) and (23), we obtain

$$P_v v_a + P_a = Q_v v_c + Q_0$$  \hspace{1cm} (24)

17. Classical Method for Max $\int_0^\infty F(x, y) \, dt$

The problem discussed in Section 15 may be set up and solved using the classical calculus of variations. We define a new function

$$H(x, y, x') = F(x, y) + \lambda(t) [x' - G(x, y)]$$  \hspace{1cm} (1)

where $\lambda$ is function of time. The Euler-Lagrange equations for $x$ and $y$ are

$$\frac{d\lambda(t)}{dt} - \left( \frac{\partial F}{\partial x} - \lambda(t) \frac{\partial G}{\partial x} \right) = 0$$  \hspace{1cm} (2)

$$\frac{d\lambda(t)}{dt} - \left( \frac{\partial F}{\partial y} - \lambda(t) \frac{\partial G}{\partial y} \right) = 0$$  \hspace{1cm} (3)

The boundary conditions are

$$x(0) = \epsilon, \quad t = 0$$  \hspace{1cm} (4)

$$\frac{\partial H}{\partial x'} = 0, \quad t = \infty$$  \hspace{1cm} (5)

Since $\partial H/\partial x' = 0$ at $t = \infty$

$$\lambda(\infty) = 0$$  \hspace{1cm} (6)

Since Eqs. (2) and (3) do not contain the independent variable $t$ explicitly, we may write immediately the first integral to (2) and (3):

$$F(x, y) + \lambda(t)[x' - G(x, y)] - \lambda(t)x' = C_1$$  \hspace{1cm} (7)

$$F(x, y) + \lambda(t)[x' - G(x, y)] = C_2$$  \hspace{1cm} (8)
where $C_1$ and $C_2$ are constants. Subtracting (7) from (8)

$$
\lambda(t) x' = C_2 - C_1
$$

(9)

But at $t = \infty$, $\lambda(\infty) = 0$. Therefore

$$
C_1 = C_2
$$

(10)

We now define a new function

$$
\mathcal{F}(x, y) = F(x, y) - C_1
$$

(11)

It follows that

$$
\frac{\partial F}{\partial y} = \frac{\partial \mathcal{F}}{\partial y}
$$

(12)

$$
\frac{\partial F}{\partial x} = \frac{\partial \mathcal{F}}{\partial x}
$$

(13)

The first integrals (7) and (8) are now written as

$$
\mathcal{F}(x, y) - \lambda(t) G(x, y) = 0
$$

(14)

$$
\mathcal{F}(x, y) = 0
$$

(15)

From (3), we express $\lambda(t)$ as

$$
\lambda(t) = \frac{\partial F/\partial y}{\partial G/\partial y} = \frac{\partial \mathcal{F}/\partial y}{\partial G/\partial y}
$$

(16)

Substituting (16) in (14) we get

$$
\mathcal{F}(x, y) - \frac{\partial \mathcal{F}/\partial y}{\partial G/\partial y} G(x, y) = 0
$$

(17)

$$
\mathcal{F}(x, y) \frac{\partial G(x, y)}{\partial y} - G(x, y) \frac{\partial \mathcal{F}(x, y)}{\partial y} = 0
$$

(18)

At the initial conditions, (18) becomes

$$
\mathcal{F}(c, v) G_v(c, v) - G(c, v) \mathcal{F}_v(c, v) = 0
$$

(19)

This equation is of the same form as that found by dynamic programming [see Section 15, Eq. (13)].

If we compare Eq. (16) of this section with Eq. (11) of Section 15, we see that the Lagrangian multiplier at time $t = 0$, namely, $\lambda(0)$, plays the same role as $-f'(c)$.
18. Euler-Lagrange Equation Derivation by Dynamic Programming

A topic of great importance is whether the nonlinear partial differential equation developed by the dynamic programming approach leads to the same results as the Euler-Lagrange equations. In other words, are these two radically different approaches really equivalent?

If we are given that

\[ J(y) = \int_0^T F(y, y', t) \, dt \]  

and the initial condition

\[ y(0) = c \]

then for fixed \( T \) we may imbed the problem into a larger class of problems over the range \((a, T)\), where \( a \) lies in the region \( 0 < a < T \).

We define

\[ f(c, a) = \min_y J(y) = \min_y \int_a^T F(y, y', t) \, dt \]  

where

\[ y(a) = c \]

Following a procedure given in Sections 13–16, we show that

\[ f(c, a) = \min_{y[a,a+S]} \left[ \min_{y[a+S,T]} \left[ \int_a^{a+S} F(y, y', t) \, dt + \int_{a+S}^T F(y, y', t) \, dt \right] \right] \]  

\[ f(c, a) = \min_{y[a,a+S]} \left[ \int_a^{a+S} F(y, y', t) \, dt + \min_{y[a+S,T]} \int_{a+S}^T F(y, y', t) \, dt \right] \]

For small \( S \), Eq. (6) becomes

\[ f(c, a) = \min_{y[a,a+S]} \left[ F(y, y', t) \, S + f(c + y' S, a + S) \right] \]

Expanding in a Taylor series, we have

\[ f(c, a) = \min_{y[a,a+S]} \left[ F(y, y', t) \, S + f(c, a) + \frac{\partial f}{\partial c} y' S + \frac{\partial f}{\partial a} S \right] \]

Letting \( S \to 0 \), we have

\[ 0 = \min_{y'} [F(c, y', a) + y' f_c + f_a] \]
where
\[ y'(a) = v' \]  (10)

We observe that as \( S \to 0 \)
\[ \frac{y(a) - y(a + S)}{S} = y'(a) = v' \]  (11)

Since \( y(a) = c \) is known and since the interval \( S \) may be considered known, the choosing of \( y \) in the range \( (a, a + S) \) is equivalent to choosing \( y'(a) \). Therefore, in (9), the minimization is taken over \( v' \).

Taking the derivative of (9) with respect to \( v' \) yields
\[ F_{v'} + f_e = 0 \]  (12)

At the minimum of (9), the following equation holds:
\[ F(c, v', a) + v'f_e + f_a = 0 \]  (13)

Taking the total derivative of (12) with respect to \( a \) and the partial derivative of (13) with respect to \( c \) we have\(^\dagger\)
\begin{align*}
\frac{d}{da} F_{v'} + f_{ae} + f_{cc}v' &= 0 \quad (14) \\
F_c + v'f_{cc} + f_{ae} &= 0 \quad (15)
\end{align*}

Subtracting (15) from (14), we have
\[ \frac{d}{da} F_{v'} - F_c = 0 \]  (16)

This is the Euler-Lagrange equation. It may be put into a more familiar form if we refer to (1) and associates \( a \) with \( t \), \( v' \) with \( y' \), and \( c \) with \( y \). We then have
\[ \frac{d}{dt} \frac{\partial F}{\partial y'} - \frac{\partial F}{\partial y} = 0 \]  (17)

\[^\dagger\quad \frac{d}{da} \frac{\partial f}{\partial c} = \frac{\partial}{\partial c} \frac{df(a, c)}{da} = \frac{\partial}{\partial c} \left[ \frac{\partial f}{\partial a} + \frac{\partial f}{\partial c} \frac{dc}{da} \right] \]
\[ \frac{d}{da} \frac{\partial f}{\partial c} = f_{ac} + f_{cc} \frac{dc}{da} \]

By (4), (10), and (11) it follows that
\[ \frac{dc}{da} = v' \]

therefore
\[ \frac{d}{da} \frac{\partial f}{\partial c} = f_{ac} + f_{cc}v' \]
We have shown, therefore, that the nonlinear partial differential equation (9), developed from the dynamic programming point of view, is equivalent to the Euler-Lagrange equation.

19. Numerical Evaluation of $f_T = \max_v \left[ F(c, v) + G(c, v) f_c \right]$

The numerical evaluation of Eq. (19), Section 14,

$$\frac{\partial f(c, T)}{\partial T} = \max_v \left[ F(c, v) + G(c, v) \frac{\partial f(c, T)}{\partial c} \right]$$  \hspace{1cm} (1)

offers a way to solve the equation when analytical solutions are difficult to obtain. The numerical solution to this type of equation is, however, not trivial. In fact, one seeks other means to solve the problem setup in Eqs. (1)–(3) of Section 14.† A sketch of the computational procedure is now given. Since the function $f$ is dependent on two variables, $c$ and $T$, we must solve this problem in two dimensions $c$ and $T$.

First, we employ the following finite difference approximations:

$$\frac{\partial f(c, T)}{\partial T} \approx \frac{f(c, T + \Delta) - f(c, T)}{\Delta}$$  \hspace{1cm} (2)

$$\frac{\partial f(c, T)}{\partial c} \approx \frac{f(c + \delta, T) - f(c, T)}{\delta}$$  \hspace{1cm} (3)

where $\Delta$ and $\delta$ are the increments in $T$ and $c$, respectively.

We substitute (2) and (3) into (1)

$$\frac{f(c, T + \Delta) - f(c, T)}{\Delta} = \max_v \left[ F(c, v) + G(c, v) \left\{ \frac{f(c + \delta, T) - f(c, T)}{\delta} \right\} \right]$$  \hspace{1cm} (4)

$$f(c, T + \Delta) = f(c, T) + \max_v \Delta \left[ F(c, v) + G(c, v) \left\{ \frac{f(c + \delta, T) - f(c, T)}{\delta} \right\} \right]$$  \hspace{1cm} (5)

\[ c = 0, 1\delta, 2\delta, \ldots; \quad T = 0, 1\Delta, 2\Delta, \ldots \]

† Rather than deal with Eq. (19) of Section 14, an alternate method for determining $c$ and $v$ is to return to Eq. (11) of Section 14. One equivalent form of Eq. (11), Section 14, is

$$f(c, T + \Delta) = \max_v \left[ F(c, v) \Delta + f(c + G(c, v) \Delta, T) \right]$$

This method was used by Bellman and Dreyfus [27].
To secure an initial condition for Eq. (5), we refer to Eq. (4) of Section 14. If $T$ is very small, this equation may be written as

$$f(c, T) = \max_{v \in \{0, T\}} [F(c, v) T]$$

(6)

In particular, at $T = 0$, we have

$$f(c, 0) = 0$$

(7)

This supplies the very necessary initial condition for solving Eq. (5).

We will now set up a grid in $\Delta$ and $\delta$ and show how to develop the function $f(c, T)$ at each node of the grid in Fig. 4.

As a consequence of (7), $f(c, 0) = 0$ for all values of $c$, $c = 0, 1\delta, 2\delta$, etc. This boundary condition is the key to the numerical solution of the partial differential equation. As a consequence of this boundary condition, we observe further that at $T = 0$ the partial derivative

$$\frac{\partial f(c, 0)}{\partial c} \approx \frac{f(c + \delta, 0) - f(c, 0)}{\delta} = 0, \quad c = 0, 1\delta, 2\delta, \ldots$$

(8)

From Eq. (5) for $T = 0$

$$f(c, 1\Delta) = f(c, 0) + \max \Delta \left[ F(c, v) + G(c, v) \left\{ \frac{f(c + \delta, 0) - f(c, 0)}{\delta} \right\} \right]$$

(9)
This equation evaluates $f(c, 1\Delta)$ along the horizontal line $1\Delta$ for $c = 0, 1\delta, 2\delta, \text{etc.}$ To carry out the details for $c = 0, T = 0$:

$$f(0, 1\Delta) = f(0, 0) + \max_v \Delta \left[ F(0, v) + G(0, v) \left\{ \frac{f(1\delta, 0) - f(0, 0)}{\delta} \right\} \right]$$

(10)

But

$$f(\delta, 0) = 0$$

(11)

$$f(0, 0) = 0$$

(12)

by the initial boundary condition $f(c, 0) = 0$ for all $c$:

$$f(0, 1\Delta) = \max_v \Delta F(0, v)$$

(13)

Continuing, if $c = 1\delta$, and $T = 0$, we have by Eq. (5)

$$f(1\delta, 1\Delta) = f(1\delta, 0) + \max_v \Delta \left[ F(1\delta, v) + G(1\delta, v) \left\{ \frac{f(2\delta, 0) - f(1\delta, 0)}{\delta} \right\} \right]$$

(14)

By virtue of Eq. (7), we write

$$f(1\delta, 1\Delta) = \max_v \Delta F(1\delta, v)$$

(15)

Following this procedure, we may write the general expression

$$f(k\delta, 1\Delta) = \max_v \Delta F(k\delta, v)$$

(16)

This gives the value of the function for the nodes along the horizontal line $1\Delta$ in Fig. 4.

To evaluate the function along the line $2\Delta$, we start with $c = 0, T = 1\Delta$ in Eq. (5):

$$f(0, 2\Delta) = f(0, 1\Delta) + \max_v \Delta \left[ F(0, v) + G(0, v) \left\{ \frac{f(1\delta, 1\Delta) - f(0, 1\Delta)}{\delta} \right\} \right]$$

(17)

But, by Eqs. (13) and (15), respectively,

$$f(0, 1\Delta) = \max_v \Delta F(0, v)$$

(18)

$$f(1\delta, 1\Delta) = \max_v \Delta F(1\delta, v)$$

(19)

$$f(0, 2\Delta) = \max_v \Delta F(0, v)$$

$$+ \max_v \Delta \left[ F(0, v) + G(0, v) \left\{ \frac{\max_v \Delta [F(1\delta, v) - F(0, v)]}{\delta} \right\} \right]$$

(20)
Continuing for \( c = 1 \delta, \, T = 1 \Delta \), we have

\[
f(1\delta, 2\Delta) = f(1\delta, 1\Delta) + \max_v \frac{\Delta}{\delta} \left[ F(1\delta, v) + G(1\delta, v) \right] \frac{f(2\delta, 1\Delta) - f(1\delta, 1\Delta)}{\delta}
\]

(21)

But, by Eqs. (15) and (16), respectively,

\[
f(1\delta, 1\Delta) = \max_v \Delta F(1\delta, v) \tag{22}
\]

\[
f(2\delta, 1\Delta) = \max_v \Delta F(2\delta, v) \tag{23}
\]

\[
f(1\delta, 2\Delta) = \max_v \Delta F(1\delta, v) + \max_v \Delta \left[ F(1\delta, v) + G(1\delta, v) \right] \frac{\max_v \Delta [F(2\delta, v) - F(1\delta, v)]}{\delta}
\]

(24)

The general expression for \( c = k\delta, \, T = 1\Delta \) is

\[
f(k\delta, 1\Delta) = \max_v \Delta F(k\delta, v) + \max_v \Delta \left[ F(k\delta, v) + G(k\delta, v) \right] \frac{\max_v \Delta [F((k-1)\delta, v) - F((k-1)\delta, v)]}{\delta}
\]

(25)

It is important to recognize that the solution is generated by evaluating \( f(c, \, T) \) at constant \( T \) for various values of \( c \). In other words, the function is evaluated by moving horizontally across the grid. The solution development along a horizontal line depends upon values developed from lines below.

If one tries to evaluate the equation at fixed \( c \) and variable \( T \), (by vertical lines), the solution is stymied. This is due to the fact that numerical values are required for some functions which are not developed in the proper sequence to be useful. For example, for \( c = 0, \, T = 0 \),

\[
f(0, 1\Delta) = f(0, 0) + \max_v \Delta \left[ F(0, v) + G(0, v) \right] \frac{f(1\delta, 0) - f(0, 0)}{\delta}
\]

(26)

\[
f(0, 1\Delta) = \max_v \Delta F(0, v)
\]

(27)

which have been determined before as Eqs. (10) and (13).

Let us continue for \( c = 0 \) and \( T = 1\Delta \)

\[
f(0, 2\Delta) = f(0, 1\Delta) + \max_v \Delta \left[ F(0, v) + G(0, v) \right] \frac{f(1\delta, 1\Delta) - f(0, 1\Delta)}{\delta}
\]

(28)
Here \( f(0, \Delta t) \) has been found in Eq. (27) but \( f(1\Delta t, \Delta t) \) has not been determined by this sequence. Since \( f(1\Delta t, \Delta t) \) is not known, Eq. (28) cannot be solved and the numerical evaluation comes to a halt.

Comments

As in the numerical solution of all differential and partial differential equations, the solution discussed here is beset by stability and convergence problems. The proper grid size in \( \delta \) and \( \Delta \) and the proper approximating forms to the partial derivatives must be carefully studied for each problem.

It is interesting to note that the dynamic programming approach leads to a partial differential equation, which is an initial value problem. The classical calculus of variations approach leads to the Euler-Lagrange equation which is a two-point boundary value problem. Generally speaking, boundary value problems are more difficult to solve than initial value problems.

20. Alternate Approach. Finite Analog to \( \max \int_0^T F(x, y) \, dt \)

Rather than deal with the partial differential equation (19), Section 14, to solve the original problem given in (1)–(3) of Section 14, we may use another approach. This approach is to use directly the defining equations (1)–(3) of Section 14, or rather discrete approximations to them.\(^\dagger\)

We break the time span \( (0, T) \) into \( N \) equal increments of length \( \Delta \), so \( N\Delta = T \). We count time backward so that \( N = N \) refers to the beginning of the process with \( N \) stages remaining, and \( N = 1 \) refers to the last stage, with one stage remaining.

At the \( i \)th time increment, we employ the following notation:

\[
x_i = x(i\Delta) \\
y_i = y(i\Delta)
\]

The derivative equation (2) of Section 14 is approximated by

\[
x_{i-1} = x_i + \Delta G(x_i, y_i), \quad i = 1, 2, \ldots, N
\]

The initial condition in Section 14, \( x(0) = c \) becomes here

\[
x_N = c
\]

\(^\dagger\) See also footnote to Section 19.
The initial value of $y$ is

$$y_N = v$$  \hspace{1cm} (5)$$

We may now approximate the integral of Section 14 by a summation

$$\sum_{i=1}^{N} F(x_i, y_i) \Delta \approx \int_{0}^{T} F(x, y) \, dt$$  \hspace{1cm} (6)$$

Let us assume that $y_i$ is constrained as

$$0 \leq y_i \leq x_i, \quad i = 1, 2, ..., N$$  \hspace{1cm} (7)$$

We define

$$f_N(c) = \max_{y_i} \Delta \sum_{i=1}^{N} F(x_i, y_i)$$  \hspace{1cm} (8)$$

Using the Principle of Optimality, we have

$$f_N(c) = \max_{0 \leq y_N \leq x_N} [\Delta F(c, y_N) + f_{N-1}(c + \Delta G(c, y_N))]$$  \hspace{1cm} (9)$$

or

$$f_N(c) = \max_{0 \leq y \leq c} [\Delta F(c, \upsilon) + f_{N-1}(c + \Delta G(c, \upsilon))]$$  \hspace{1cm} (10)$$

$$f_i(c) = \max_{0 \leq \upsilon \leq c} \Delta F(c, \upsilon)$$  \hspace{1cm} (11)$$

Figure 5 illustrates the terms in Eq. (10).
21. Finite Analog to Min $\int_0^T F(y, y', t) \, dt$

Since this form is used so often as a point of departure in the calculus of variations, it will be interesting to set up the finite analog by dynamic programming.

The problem is to minimize $J(y')$, where

$$J(y') = \int_0^T F(y, y', t) \, dt$$

(1)

with the initial condition

$$y(0) = c$$

(2)

We break the time span (0, $T$) into $N$ equal increments of $\Delta$ duration, so $N\Delta = T$. We count time forward so $N = 0$ represents the beginning of the process and $N = N$ represents the end of the process.

We approximate Eq. (1) by

$$J(y_k) = \Delta \sum_{k=0}^{N-1} F(y_k, y_k', k)$$

(3)

where

$$y_k = y(k\Delta)$$

(4)

$$\frac{y_{k+1} - y_k}{\Delta} = y_k', \quad k = 0, 1, 2, ..., N$$

(5)

$$y_0 = c$$

(6)

If we consider $N$ to be known and fixed, we may imbed Eq. (3) into a larger class of problems by summing from $k = a$, to $k = N - 1$, where $a = 0, 1, 2, ..., N - 1$. In this problem, we count time forward so that $a$ is the beginning of the process and $N$ is the end of the process.

We now define

$$f_a(c) = \text{the minimum of } J(y_k') \text{ over } y_k' \text{ starting in stage } a \text{ and continuing to stage } (N - 1), \text{ beginning in state } y_a = c \text{ and pursuing an optimal policy}$$

$$f_a(c) = \text{Min } \Delta \sum_{k=a}^{N-1} F(y_k, y_k', k), \quad a = 0, 1, 2, ..., N - 1$$

(7)

The initial condition is

$$y_a = c$$

(8)

Equation (7) may be broken up into two parts; one for the immediate return and the other for the ($a + 1$) remaining stages.
The functional equations are

\[ f_n(c) = \min_{y_k} [\Delta F(c, y_a, a) + f_{n+1}(c + y_a \Delta)] \] (9)

\[ f_N(c) = 0 \] (10)

The last two equations provide a working algorithm.

## 22. Control of Chemical Reactor

The formulation above in Section 14 may be extended to a more general problem, namely, maximizing

\[ J(z) = \int_0^T F(x_1, x_2, ..., x_n, z) \, dt \] (1)

subject to the differential equations

\[ \frac{dx_i}{dt} = G_i(x_1, x_2, ..., x_n, z), \quad i = 1, 2, ..., n \] (2)

\[ x_i(0) = c_i \] (3)

and subject to the general constraints

\[ R_k(x_1, x_2, ..., x_n, z) \leq 0, \quad k = 1, 2, ... \] (4)

In the formulation \( z(t) \) is the control variable that must be chosen to maximize \( J(z) \) over the region \((0, T)\).

In a chemical reaction, we might think of the \( x_i \) terms as the concentrations of the reacting chemicals, the \( z(t) \) might be the temperature to be controlled or manipulated. The \( G_i \) terms represent the rate of reaction for each component. The \( R_k \) terms may be bounds on the permissible ranges for compositions and temperatures, perhaps

\[ (x_i)_{\text{Min}} \leq x_i \leq (x_i)_{\text{Max}} \] (5)

\[ (z_i)_{\text{Min}} \leq z_i \leq (z_i)_{\text{Max}} \] (6)

The \( R_k \) could also encompass more complicated expressions such as

\[ Ax_i^2 + Bz_i \leq C \] (7)

where \( A, B, \) and \( C \) are known constants.
To illustrate the development of expressions similar to those in Section 14, let us, after Aris [1], consider the reaction

$$X \rightarrow Y \rightarrow Z$$

where $Y$ is desired and $Z$ is a waste product.

The reaction equations are

$$\frac{dx}{dt} = -a(\theta)F(x)$$  \hspace{1cm} (8)

$$\frac{dy}{dt} = na(\theta)F(x) - b(\theta)G(y)$$  \hspace{1cm} (9)

where $a(\theta)$ and $b(\theta)$ are the reaction velocity constants, which are a function of the reaction temperature $\theta$, where $x$, $y$ are the concentrations of $X$ and $Y$, and where $n$ is a known constant.

We desire to maximize the yield of $Y$ over the holding time in the reactor. We let

\[ t = \text{the holding time in the reactor up to a point} \]
\[ T = \text{the total holding time in the reactor} \]
\[ 0 < t < T \]

In other words, we desire to maximize $J(\theta)$

$$J(\theta) = \int_{0}^{T} \frac{dy}{dt} dt = y(T) - y(0)$$  \hspace{1cm} (10)

$$J(\theta) = \int_{0}^{T} \{na(\theta)F(x) - b(\theta)G(y)\} \, dt$$  \hspace{1cm} (11)

We define

$$f(x_0, y_0, T) = \text{Max } J(\theta) = \text{the maximum yield of } Y, \text{ starting with the initial quantities of } X \text{ and } Y, x_0 \text{ and } y_0, \text{ and a total holding time } T, \text{ and pursuing an optimal policy}$$

$$f(x_0, y_0, T) = \text{Max } \int_{0}^{T} \{na(\theta)F(x) - b(\theta)G(y)\} \, dt$$  \hspace{1cm} (12)

Proceeding as in section 14, we have

$$\frac{\partial f}{\partial T} = \text{Max } \left[ (naF - bG)_0 + \frac{\partial f}{\partial x_0} \left( \frac{dx_0}{dt} \right) + \frac{\partial f}{\partial y_0} \left( \frac{dy_0}{dt} \right) \right]$$  \hspace{1cm} (13)

where the subscript zero refers to initial conditions.

We may now substitute Eqs. (8) and (9) into (13)

$$\frac{\partial f}{\partial T} = \text{Max } \left[ (naF - bG)_0 \left( 1 + \frac{\partial f}{\partial y_0} \right) - (aF)_0 \frac{\partial f}{\partial x_0} \right]$$  \hspace{1cm} (14)
If we suppose for the moment that the process is not constrained by the equations $R_k$, then at the maximum, the two following equations are required:

$$ (n\alpha'(\theta_0)F(x_0) - b'(\theta_0) G(y_0)) \left(1 + \frac{\partial f}{\partial y_0}\right) - a'(\theta_0) F(x_0) \frac{\partial f}{\partial x_0} = 0 \quad (15) $$

$$ \frac{\partial f}{\partial T} = (n\alpha(\theta_0)F(x_0) - b(\theta_0) G(y_0)) \left(1 + \frac{\partial f}{\partial y_0}\right) - a(\theta_0) F(x_0) \frac{\partial f}{\partial x_0} \quad (16) $$

The solution of these yields the $x$ and $y$ time functions that maximize the $Y$ production.

Method of Characteristics

In order to discuss the solution of Eqs. (15) and (16), let us first review briefly the method of characteristics. One technique for solving partial differential equations is to set up the characteristic equations of the partial differential equation. These equations are ordinary differential equations, the solution of which yields the solution to the original partial differential equation. The advantage of the method of characteristics is that it generates a set of ordinary differential equations which are easier to solve than the original partial differential equation. In some problems, the characteristic equations give a deeper insight into the physical significance of the equations. Let us consider the linear partial differential equation:

$$ L = L(f_{x_0}, f_{y_0}, f_T, f, x_0, y_0, T) = 0 \quad (17) $$

where

$$ f = f(x_0, y_0, T) \quad (18) $$

$$ f_{x_0} = \frac{\partial f}{\partial x_0} $$

$$ f_{y_0} = \frac{\partial f}{\partial y_0} \quad (19) $$

$$ f_T = \frac{\partial f}{\partial T} $$

Then the characteristic equations are

$$ \frac{dx_0}{dt} = \frac{\partial L}{\partial f_{x_0}} $$

$$ \frac{dy_0}{dt} = \frac{\partial L}{\partial f_{y_0}} $$
Application of the Method of Characteristics

Returning to Eqs. (15) and (16), we may combine these to form a single partial differential equation:

\[
H = H(f_x, f_y, f_T, x_0, y_0) = 0
\]  

The partial differential equation \( H \) differs from the partial differential equation \( L \) in Eq. (17) only in the absence of \( f \) and \( T \). Since \( \partial H / \partial f \) and \( \partial H / \partial T \) are identically zero, the corresponding characteristic equations are

\[
\frac{dx_0}{dt} = \frac{\partial H}{\partial f_{x_0}}
\]
\[
\frac{dy_0}{dt} = \frac{\partial H}{\partial f_{y_0}}
\]
\[
\frac{dT}{dt} = \frac{\partial H}{\partial f_T}
\]
\[
\frac{df}{dt} = f_{x_0} \frac{\partial H}{\partial f_{x_0}} + f_{y_0} \frac{\partial H}{\partial f_{y_0}} + f_T \frac{\partial H}{\partial f_T}
\]  

\[
\frac{df_{x_0}}{dt} = - \frac{\partial H}{\partial x_0}
\]
\[
\frac{df_{y_0}}{dt} = - \frac{\partial H}{\partial y_0}
\]
\[
\frac{df_T}{dt} = 0
\]
So far we have considered the solution to the partial differential equation (14) if it is unconstrained. If it is constrained so Eq. (4) applies, then the characteristic equations are solved as unconstrained problems until a bound is reached. From then on the characteristic equations are solved with the constrained variable held at its bound [1, 2].

For the unconstrained case, another approach is to evaluate Eq. (14) by numerical means as discussed in Section 19.

The characteristics offer a convenient way to solve the partial differential equation. In effect, the method of characteristics replaces a single partial differential equation by a set of ordinary differential equations. In the example given above, the characteristic equations yield directly the reaction paths, that is, \( x \) and \( y \) as a function of time.

23. Example: Smoothing Problems as Variational Problems

Smoothing problems are an interesting class of variational problems occurring in industry. These problems are characterized by production and inventory processes in which it is desired to modulate fluctuations in the manufacture and inventory of the products. The "smoothing" operation rounds out, so to speak, large fluctuations in the process behavior.

A general expression for the smoothing problem is to minimize the functional

\[
J(x) = \int_0^T [F_1(x - b_1(t), t) + F_2(x' - b_2(t), t) + \ldots + F_k(x^{(k-1)} - b_k(t), t)] \, dt
\]

subject to the initial conditions

\[
x(0) = c_1, \quad x'(0) = c_2, \ldots, x^{(k-1)}(0) = c_k
\]

Let us consider the following simple form:

\[
J(x) = \int_0^T [F_1(x - b_1(t), t) + F_2(x' - b_2(t), t)] \, dt
\]

We may consider \( x(t) \) to be the production rate, \( b(t) \) to be the desired product demand rate, \( x'(t) \) to be the rate of change in the production rate, and \( b_2(t) \), the permitted average rate of change in production rate. The function \( F_1 \) is the cost of failing to meet the production demand and the function \( F_2 \) is the cost for exceeding the permitted average rate of change in production. It is desired to minimize these total costs.
While other interpretations can be given to the various terms in Eqs. (1) and (3), the interplay involved in satisfying multiple conflicting demands over the time interval \((0, T)\) makes for a challenging problem.

Let us imbed the problem of Eq. (3) into a larger class of problems by considering the interval \((a, T)\) where \(0 \leq a \leq T\), and where \(T\) is fixed. Let us also consider the case where \(F_1\) and \(F_2\) are quadratic. In particular, we take

\[
F_1[x - b_1(t), t] = K_1(t) (x - b_1(t))^2
\]

\[
F_2[x' - b_2(t), t] = K_2(t) (x' - b_2(t))^2
\]

The problem then consists of minimizing

\[
J(a, x) = \int_a^T [K_1(t) (x - b_1(t))^2 + K_2(t) (x' - b_2(t))^2] \, dt
\]

Let us define

\[
f(a, c) = \min_x J(a, x)
\]

where

\[
x(a) = c
\]

\[
x' = \frac{dx}{dt} = G(a, c) = \text{a function to be determined}
\]

\[
K_1(t), K_2(t) \geq 0, \quad t \geq 0
\]

Proceeding as before, we develop

\[
0 = \min_{G(a,c)} \left[ K_1(a) (c - b_1(a))^2 + K_2(a) (G(a,c) - b_2(a))^2 + \frac{\partial f(a,c)}{\partial a} + G(a,c) \frac{\partial f(a,c)}{\partial c} \right]
\]

Here, the minimization is over \(G(a, c)\) rather than \(x\) in the region \((a, a + S)\).

Since \(x(a) = c\) is known and \(G(a, c)\) is not known, the minimization of Eq. (7) by choosing the proper value of \(x\) over the interval \((a, a + S)\) where \(S \to 0\) is equivalent to choosing the proper value of \(G\) at time \(a\); that is to say,

\[
G(a,c) = \lim_{S \to 0} \frac{x(a) - x(a + S)}{S}
\]
Taking the derivative of Eq. (11) with respect to \( G(a, c) \), we find

\[
2K_2(a) [G(a, c) - b_2(a)] + \frac{\partial f(a, c)}{\partial c} = 0
\]

(13)

Solving Eq. (11) for \( \frac{\partial f(a, c)}{\partial a} \), we have

\[
\frac{\partial f(a, c)}{\partial a} = -K_1(a) [c - b_1(a)]^2 - K_2(a) [G(a, c) - b_2(a)]^2 - G \frac{\partial f(a, c)}{\partial c}
\]

(14)

Equations (13) and (14) are equivalent to Eq. (11). Equation (14) can be simplified by eliminating \( G(a, c) \) between Eqs. (13) and (14)

\[
f_a = -K_1(a) [c - b_1(a)]^2 - b_2(a)f_c + \frac{f_c^2}{4K_2(a)}
\]

(15)

Equation (11) determines \( G(a, c) \) if \( f(a, c) \) is known. The initial value for \( f(T, c) \) is

\[
f(T, c) = 0
\]

(16)

One possible way to solve Eq. (11) is by numerical means. Another way is to take advantage of the analytical form for \( f(a, c) \) and equate coefficients, which we do below.

Let us assume \( f(a, c) \) takes the form

\[
f(a, c) = u(a) + cv(a) + c^2w(a)
\]

(17)

where \( u(a) \), \( v(a) \), and \( w(a) \) are to be determined.

Forming \( \frac{\partial f}{\partial a} \) and \( \frac{\partial f}{\partial c} \) from Eq. (17), we substitute into Eq. (15)

\[
u'(a) + cv'(a) + c^2w'(a) = -K_1(a) [c - b_1(a)]^2 - b_2(a)[v(a) + 2cw(a)]
+ \frac{[v(a) + 2cw(a)]^2}{4K_2(a)}
\]

(18)

\[
u'(a) + cv'(a) + c^2w'(a) = -K_1(a) b_1^2(a) - b_2(a)[v(a) + \frac{v^2(a)}{4K_2(a)}]
+ c \left[ 2b_1(a)K_1(a) - 2b_2(a)w(a) + \frac{v(a)w(a)}{K_2(a)} \right]
+ c^2 \left[ -K_1(a) + \frac{w^2(a)}{K_2(a)} \right]
\]

(19)
Collecting coefficients of \( c^0, c^1, c^2 \), we have the differential equations

\[ u'(a) = -K_1(a) b_2^2(a) - b_3(a) v(a) + \frac{v^2(a)}{4K_3(a)} \]  
\[ v'(a) = 2b_1(a) K_1(a) - 2b_3(a) w(a) + \frac{v(a) w(a)}{K_3(a)} \]  
\[ w'(a) = -K_1(a) + \frac{v^2(a)}{K_3(a)} \]

Consistent with the initial value condition in Eq. (16), the initial conditions for Eqs. (20) to (22) are

\[ u(T) = v(T) = w(T) = 0 \]  

In the set of equations (20) to (22) notice that once \( w(a) \) has been found from Eq. (22), it can then be substituted into Eq. (21) which can be solved for \( v(a) \). Knowledge of \( v(a) \) permits \( u(a) \) in Eq. (20) to be found. Equation (22) is a Riccati equation which can be solved in terms of the solution of linear equations [37, 41].

As a consequence of the solution of Eqs. (20)–(22), the function \( f(a, c) \) may be found from Eq. (17). The knowledge of \( f(a, c) \) permits \( G(a, c) \) to be evaluated by Eq. (13). In particular, Eq. (13) may be rewritten as

\[ 2K_3(a) [G(a, c) - b_3(a)] + v(a) + 2c w(a) = 0 \]  

or

\[ G(a, c) = -\frac{v(a) + 2c w(a)}{2K_3(a)} + b_3(a) \]

Since \( c \) is the value of \( x \) at time \( a \), Eq. (25) implies that

\[ G(t, x) = -\frac{v(t) + 2c w(t)}{2K_3(t)} + b_3(t) \]

Since \( dx/dt = G(t, x) \) by Eq. (9), \( x \) may be found explicitly as a function of \( c \) and \( t \).

24. Example: Brachistochrone

One of the first problems discussed in the calculus of variations was the brachistochrone problem. This problem deals with seeking the minimum path from a known origin to a known final position.
We will discuss here the calculus of variations formulation of this problem and then the dynamic programming formulation. An extension of this problem to the control of a batch chemical reaction will also be given.

a. Calculus of Variations

The problem may be formulated as follows. Find the minimum time of travel for a particle moving only in a vertical plane under the influence of gravity from the origin \((x_0, y_0)\) to the known final position \((x_F, y_F)\) (Fig. 6).

The velocity of the particle at any position \(y\) is

\[ v = \sqrt{2gy} \]  

(1)

and the element of distance along the curve is

\[ ds = \sqrt{(dx)^2 + (dy)^2} \]  

(2)

whence

\[ \frac{ds}{dx} = \sqrt{1 + (y')^2}, \quad \text{where} \quad y' = \frac{dy}{dx} \]  

(3)

\[ ds = \sqrt{1 + (y')^2} \, dx \]  

(4)

The time of travel is

\[ \int_0^t \, dt = \int_{x_0}^{x_F} \frac{ds}{x_0 \, v} \]  

(5)
Hence,
\[
t = \frac{1}{\sqrt{2g}} \int_{x_0}^{x_F} \sqrt{\frac{1 + (y')^2}{y}} \, dx
\]  
(6)

Our objective is to minimize \( t \).

The correspondence between Eq. (6) of the section and Eq. (1) of Section 2 is
\[
F(y, y', x) = \frac{1}{\sqrt{2g}} \sqrt{\frac{1 + (y')^2}{y}}
\]  
(7)
where \( x \) is the independent variable. The Euler-Lagrange equation corresponding to Section 2, Eq. (3) is
\[
\frac{d}{dx} \left[ \frac{y'}{\sqrt{y(1 + (y')^2)}} \right] + \frac{y^{3/2}}{2} \sqrt{1 + (y')^2} = 0
\]  
(8)
Since the function \( F \) does not contain the independent variable \( x \), we may write immediately a first integral to the Euler-Lagrange equation
\[
F - y' \frac{\partial F}{\partial y'} = K_1 = \text{constant}
\]  
(9)
or
\[
\sqrt{\frac{1 + (y')^2}{y}} - \frac{(y')^2}{\sqrt{y(1 + (y')^2)}} = K_1
\]  
(10)
whence
\[
\frac{1}{y(1 + (y')^2)} = K_1^2
\]  
(11)
Integration yields
\[
y = \frac{1 - \cos t}{2K_1^2}
\]  
(12)
\[
x = K_2 + \frac{t - \sin t}{2K_1^2}
\]  
(13)
where the boundary conditions are
\[
t = 0, \quad y = y_0 = 0, \quad x = x_0 = 0
\]
\[
t = t_F, \quad y = y_F, \quad x = y_F
\]  
(14)
b. Dynamic Programming Approach

The dynamic programming approach starts first with the expression for the time of travel, Eq. (6), which is developed as above. We define

\[ J(y) = \frac{1}{\sqrt{2g}} \int_x^{x_F} \left[ \frac{1 + (y')^2}{y} \right]^{1/2} dx \]  

(15)

the time of travel from \( x, y \) to \( x_F, y_F \)

We desire to minimize \( J(y) \) while satisfying the boundary conditions

\[ t = 0, \quad y = y_0 = 0, \quad x = x_0 = 0 \]  

(16)

\[ t = t_F, \quad y = y_F, \quad x = x_F \]

We define

\[ f(x, y) = \text{Min}_{y'} J(y) = \text{the minimum time of travel from the} \]

point \( x, y \) to the point \( (x_F, y_F) \)

(17)

Using the same approach given in Section 14, we may write

\[ f(x, y) = \text{Min}_y \left[ \sqrt{\frac{1 + (y')^2}{2gy}} \Delta x + f(x + \Delta x, y + y'\Delta x) \right] \]  

(18)

In the limit as \( \Delta x \to 0 \), we have as before

\[ 0 = \text{Min}_{y'} \left[ \sqrt{\frac{1 + (y')^2}{y}} + f_x + y'f_y \right] \]  

(19)

At the minimum, this nonlinear partial differential equation is equivalent to the two equations

\[ \frac{y'}{\sqrt{y(1 + (y')^2)}} + f_y = 0 \]  

(20)

\[ \sqrt{\frac{1 + (y')^2}{y}} + f_x + y'f_y = 0 \]  

(21)

To eliminate \( f \) between the two equations, we take the total derivative of Eq. (20) with respect to \( x \) and partial derivative of Eq. (21) with respect to \( y \):

\[ \frac{d}{dx} \left[ \frac{y'}{\sqrt{y(1 + (y')^2)}} \right] + \frac{d}{dx} \frac{\partial f}{\partial y} = 0 \]  

(22)

Since

\[ \frac{d}{dx} \frac{\partial f}{\partial y} = \frac{\partial}{\partial y} \frac{df}{dx} \]  

(23)
and since \( f \) is a function of \( x \) and \( y \), we have

\[
\frac{d}{dx} f(x, y) = \frac{\partial f}{\partial x} + \frac{\partial f}{\partial y} \frac{dy}{dx} = f_x + y' f_y
\]  

(24)

\[
\frac{\partial}{\partial y} \left( \frac{df}{dx} \right) = f_{xu} + y' f_{yu}
\]  

(25)

We now have on substituting Eq. (25) into (22)

\[
\frac{d}{dx} \left[ \frac{y'}{\sqrt{y(1 + (y')^2)}} \right] + f_{xu} + f_{yu} y' = 0
\]  

(26)

The partial derivative of Eq. (21) with respect to \( y \) is

\[
-\frac{y^{-3/2}}{2} \sqrt{1 + (y')^2} + f_{xu} + f_{yu} y' = 0
\]  

(27)

Subtracting Eq. (27) from Eq. (26), we get

\[
\frac{d}{dx} \left[ \frac{y'}{\sqrt{y(1 + (y')^2)}} \right] + \frac{y^{-3/2}}{2} \sqrt{1 + (y')^2} = 0
\]  

(28)

This equation, derived from the dynamic programming approach, is precisely the Euler-Lagrange equation given in Eq. (8).

Hence, it follows that the a first integral is given by Eq. (10) and the solution is given by Eqs. (12) and (13).

We may conclude, therefore, that the functional equation approach of dynamic programming yields a nonlinear partial differential equation, Eq. (19). The solution to Eq. (19) via Eqs. (20)-(27) leads to Eq. (28) which is identically the Euler-Lagrange equation developed by the calculus of variations approach. An alternate dynamic programming approach to evaluate \( f(x, y) \) might be to use Eq. (19) directly. This requires finding the proper value of the derivative at each point \( x, y \).

25. Batch Reaction Control, Time Minimization

Introduction

In a batch reaction, it is often desired to take a chemical system from a given known initial composition and process conditions to a final composition in a minimum time. This problem broadly falls into the "brachistochrone" class of problems, so widely discussed in the calculus of variations. In this section we discuss how to set up and solve this
problem by dynamic programming. As a consequence of this it is possible for a computer to guide the process by adjusting process variables to achieve this time minimization. We point out some of the problems as well as the utility of the method. Some of the advantages of dynamic programming over the calculus of variations are discussed.

Statement of the Problem

We consider the following batch reaction:

\[ X \xrightarrow{k_1} Y \xrightarrow{k_2} Z \]  

(1)

The equations describing this process are

\[ \frac{dx}{dt} = -k_1 x + k_2 y \]  

(2)

\[ \frac{dy}{dt} = +k_1 x - (k_2 + k_3) y \]  

(3)

\[ \frac{dz}{dt} = +k_2 y \]  

(4)

where

- \( x \) = the concentration of \( X \)
- \( y \) = the concentration of \( Y \)
- \( z \) = the concentration of \( Z \)
- \( k_i = A_i e^{-\Delta E_i/RT} \) = the reaction velocity constant, where \( A_i \) is a function of pressure
- \( \Delta E = \) activation energy
- \( R = \) universal gas constant
- \( T = \) temperature

Equation (4) is dependent on Eqs. (2) and (3), since it can be formed by the sum of Eqs. (2) and (3).

For convenience here we will consider the reactions to occur isothermally. We also consider that the reaction velocity terms \( k_i \) are functions of pressure only. This restriction is assumed here merely to simplify the numerical example to be presented. In addition we can compare the approach with that of previous results, given in references [35, 36, 38, 43]. The dynamic programming technique works with or without this restriction.

We desire to minimize the time to convert the initial composition of \( x_0, y_0, z_0 \) at pressure \( P_0 \) and temperature \( T \) to the final composition \( x_f, y_f, z_f \) at temperature \( T \).
This minimization is done by manipulating the pressure as a function of time. We observe that although the initial pressure is given, the final pressure is not specified.

For the case where the $k_i$ terms are constants, the solutions to Eqs. (2) and (3) are given by

$$x = \alpha_1 e^{-\theta t} + \alpha_2 e^{-\varphi t}$$  \hspace{1cm} (5)

$$y = \beta_1 e^{-\theta t} + \beta_2 e^{-\varphi t}$$  \hspace{1cm} (6)

where

$$\theta = \frac{k_1 + k_2 + k_3 + \sqrt{(k_1 + k_2 + k_3)^2 - 4k_1k_2}}{2}$$  \hspace{1cm} (7)

$$\varphi = \frac{k_1 + k_2 + k_3 - \sqrt{(k_1 + k_2 + k_3)^2 - 4k_1k_2}}{2}$$  \hspace{1cm} (8)

$$\alpha_1 = \frac{(\theta - k_2 - k_3)x_0 - k_3y_0}{\theta - \varphi}$$  \hspace{1cm} (9)

$$\alpha_2 = \frac{(k_2 + k_3 - \varphi)x_0 + k_3y_0}{\theta - \varphi}$$  \hspace{1cm} (10)

$$\beta_1 = \frac{-k_1(x_0 + y_0) + \theta y_0}{\theta - \varphi}$$  \hspace{1cm} (11)

$$\beta_2 = \frac{k_3(x_0 + y_0) - \varphi y_0}{\theta - \varphi}$$  \hspace{1cm} (12)

Although the solutions to Eqs. (2) and (3) are available in closed form, the equations also may be solved numerically by using finite difference approximations. These equations are

$$\frac{x_{j+1} - x_j}{\Delta t} = -k_1x_j + k_3y_j$$  \hspace{1cm} (13)

$$\frac{y_{j+1} - y_j}{\Delta t} = +k_1x_j - (k_2 + k_3)y_j$$  \hspace{1cm} (14)

$$\frac{z_{j+1} - z_j}{\Delta t} = k_2y_j$$  \hspace{1cm} (15)

If the pressure changes during the time interval $\Delta t$, the $k_i$ constants are evaluated at the average pressure.

**Dynamic Programming Approach**

Before setting up the functional equations of dynamic programming, let us first consider the process path in the $xz$ composition plane. In this plane, we know the starting composition $x_0$, $z_0$ and the final composition $x_F$, $z_F$. We desire to pursue a path across this plane which
minimizes the time of travel between the initial and final compositions. We will superimpose a grid on the $xz$ plane and insist that all paths must be taken from node to node along the legs of the grid. We cannot employ a rectangular grid for $x$ and $z$ with elements parallel to the axes since this implies that the system can move at constant $z$ or constant $x$. Equations (2)–(4) tell us that if $x$, $y$, or $z$ are constant, the derivatives are equal to zero, which is one of the conditions for equilibrium.

We also note that compound $Y$ passes through a maximum, which is also a condition for $dy/dt$ to be equal to zero. Since we are interested in describing the transient behavior of the system on the $xz$ plane, we must employ a grid with elements nonparallel to the abscissa and ordinate.

From the nature of the chemical reaction, Eq. (1), we observe that with increasing time compound $X$ always decreases, compound $Z$ always increases, and compound $Y$ increases to a maximum and then decreases. A grid that satisfactorily represents the increase of $2$ with a decrease in $X$ is given in Fig. 7. It consists simply of two sets of parallel lines with different slopes of $dz/dx$. In the limit this grid (as well as others) will provide sufficient avenues between the initial and final point to permit the best path to be chosen.

At each node of the grid we note that there are two possible choices to move, along a path at $dz/dx = -C_1$ or along a path of $dz/dx = -C_2$. Starting at the initial point where the pressure is known, one can develop the time to traverse from one node to the next throughout the entire grid by using Eqs. (5) and (6) or Eqs. (13) and (14). This is made possible by the fact that the composition is known at each grid point and the pressure can be chosen to drive the process from one composition to the next. This means that the pressure-composition-time paths are also determined.

With the times of travel determined for each leg of the grid, it is now possible to determine the time of every path from the initial point to the final point. Our task is to find the least time path.

With these preliminaries out of the way, we may now set up the functional equations describing the process. The basis for the functional equations is Bellman's Principle of Optimality.

Keeping in mind the $xz$ grid, we define the following terms:

$$f_N(x_N, z_N, P_N) = \text{the minimum time for the system to move from the current state (}x_N, z_N, P_N\text{) over the } N \text{ remaining stages to the final composition (}x_F, z_F\text{), using an optimal policy. The } N \text{ refers to the number of stages remaining, that is, the number of nodes in the grid which must be traversed to reach the final composition. When one node away from the final composition, } N = 1; \ N = 1, 2, 3, \ldots$$
\[ \Delta t_N = \text{the time to travel from node } N \text{ to node } (N - 1) \text{ along the path } \frac{dz}{dx} = -C_1 \]  

(17)

\[ \delta t_N = \text{the time to travel from node } N \text{ to node } (N - 1) \text{ along the path } \frac{dz}{dx} = -C_2 \]  

(18)

We recognize that the time to travel from node \( N \) to node \( (N - 1) \) is the same as the time to travel from node \( (N - 1) \) to node \( N \).
The minimum time path may be described by

\[ f_N(x_N, z_N, P_N) = \min_{P_{\text{min}}} \Delta t_N + f_{N-1}[x_N + \Delta t_N(-k_1 x_N + k_2 y_N), \]
\[ z_N + \Delta t_N k_3 y_N, P_{N-1}] \]
\[ \delta t_N + f_{N-1}[x_N + \delta t_N(-k_1 x_N + k_2 y_N), \]
\[ z_N + \delta t_N k_3 y_N, P_{N-1}] \]

(19)

\[ f_1(x_1, y_1, P_1) = \min_{P_{\text{min}}} \Delta t_1 \]
\[ \delta t_1 \]

(20)

The functional equations (19) and (20) describe the minimum time path using the Principle of Optimality. Equation (19) requires that the smaller of two alternatives be chosen by suitable choice of the pressure within the permitted range. The upper line of Eq. (19) determines the total time over the \( N \) remaining stages when the choice is to move along a path of \( dz/dx = -C_1 \) during the \( N \)th stage. The lower line of Eq. (19) determines the total time over the \( N \) remaining stages when the choice is to move along the path \( dz/dx = -C_3 \) during the \( N \)th stage. The first term of the upper line, \( \Delta t_N \), represents the time to pass along a path of \( dz/dx = -C_1 \) from node \( N \) to node \( (N-1) \). The expression

\[ f_{N-1}[x_N + \Delta t_N(-k_1 x_N + k_2 y_N), z_N + \Delta t_N k_3 y_N, P_{N-1}] \]

represents the minimum time to pass from node \( (N-1) \) to the final composition, beginning with the new value of \( x = x_N + \Delta t_N(-k_1 x_N + k_2 y_N) \), the new value of \( z = z_N + \Delta t_N k_3 y_N \), the new value of pressure \( P_{N-1} \) and pursuing an optimal policy. Similar statements may be made about the lower line of Eq. (19).

Equation (20) describes the one-stage process. Once Eq. (20) is evaluated, the 2-stage, 3-stage, etc., processes may be evaluated by the recursion relationship Eq. (19).

Referring to Fig. 7, we note that node \( b \) and node \( e \) are 1 stage from \( a \), the final point. We also note that nodes \( c \), \( d \), and \( f \) are 2 stages from node \( a \). To determine the minimum time from \( d \) to \( a \), we must evaluate the time path \( abd \) and the time path \( aed \) and choose the smallest. Having previously determined the time to move along each leg of the grid, we can now apply Eqs. (19) and (20). The minimum time to travel from point \( b \) to point \( a \) is found by Eq. (20). In this case there is only one way to get to node \( a \) from node \( b \), so there is really no choice. A similar statement may be made for the time of travel from node \( a \) to node \( e \). To determine the time from node \( d \) to node \( a \) we apply Eq. (19), which states that we must choose the time path along \( db \) or \( de \) so that when it
is added to all the previous times (the time from \( b \) to \( a \) or from \( e \) to \( a \), respectively) the total time will be a minimum. In this manner the minimum time from any point on the grid to the final point may be determined by working from node to node through the grid.

The important point to remember about the dynamic programming method is that it eliminates the necessity of examining every path from the node \( N \) to the final point to find the least time path. By the bootstrap operation just described, the cumulative minimum time is found by progressing one stage at a time. At node \( N \) only one decision needs to be made, to move along a path of \( dz/dx = -C_1 \), or \( dz/dx = -C_2 \), since the minimum time has already been found for \((N-1)\) nodes.

The ideas presented here will be made more clear by the following numerical example.

**Numerical Example**

A numerical example has been worked out using the following constants and conditions expressed in consistent units:

\[
\begin{align*}
k_1 &= 1.0000 \, P^{-5}, \quad k_2 = 5(10^{-4}) \, P^2, \quad k_3 = 1.2594 \\
\end{align*}
\]

<table>
<thead>
<tr>
<th>Initial Point</th>
<th>Final Point</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x_0 = 0.88000 )</td>
<td>( x_F = 0.60000 )</td>
</tr>
<tr>
<td>( y_0 = 0.11365 )</td>
<td>( y_F = 0.29765 )</td>
</tr>
<tr>
<td>( z_0 = 0.00635 )</td>
<td>( z_F = 0.10235 )</td>
</tr>
<tr>
<td>( P_0 = 298 )</td>
<td></td>
</tr>
</tbody>
</table>

A grid was employed with lines of \( \Delta z/\Delta x = -0.3 \) and \( \Delta z/\Delta x = -0.4 \).

Beginning at the initial point \( x_0, y_0, z_0 \) and at pressure \( P_0 \), the time to travel from node to node was computed by the following equations:

\[
\frac{x_{j+1} - x_j}{x_{j+1} - x_j} = \frac{k_2 y_j}{-k_1 x_j + k_3 y_j}
\]

where

\[
\frac{x_{j+1} - x_j}{x_{j+1} - x_j} = -0.3 \text{ or } -0.4
\]

\[
\Delta t_j = \frac{x_{j+1} - x_j}{-k_1 x_j + k_3 y_j}, \quad \text{along} \quad \frac{x_{j+1} - x_j}{x_{j+1} - x_j} = -0.3
\]

\[
\delta t_j = \frac{x_{j+1} - x_j}{-k_1 x_j + k_3 y_j}, \quad \text{along} \quad \frac{x_{j+1} - x_j}{x_{j+1} - x_j} = -0.4
\]

\[
x_j + y_j + z_j = 1
\]

In these equations, the composition terms \( x_j, z_j \) and \( x_{j+1}, z_{j+1} \) are known from the grid. The \( k_1 \) and \( k_2 \) terms are evaluated at the average
pressure between composition \( x_j \), \( z_j \) and \( x_{j+1} \), \( z_{j+1} \). Equation (21) is solved by selecting the average pressure from node \( j \) to node \( (j + 1) \) which yields \( k_1 \) and \( k_2 \) values that satisfy it. Having found the average pressure over the path from node \( j \) to node \( (j + 1) \), one can find the corresponding time from Eq. (22) or (23).

![Pressure-composition-time plot](image)

**Fig. 8.** Pressure-composition-time plot.

**NOTES:**
1. Drawing is not to scale. Compositions corresponding to the nodes are given in Table 1.
2. Numbers below the lines or to the left of the lines represent the time of travel between adjacent nodes in units of \( 10^{-4} \).
3. Numbers above the lines or to the right of the lines represent the average pressure to drive the composition from node \( j \) to node \( (j + 1) \), corresponding to the time interval.
Figure 8 represents the grid superimposed on the \(xz\) plane. The grid has not been drawn to scale in order to present the time and pressure data in a legible manner. The coordinates of the grid are given in Table 1. In Fig. 8 the numbers below the line or to the left of the line represent the time to travel from node \(j\) to node \((j + 1)\). The numbers above the lines or to the right of the lines represent the average pressure existing during the time interval from node \(j\) to node \((j + 1)\). Starting at the final composition we note that the minimum time of travel from \(a\) to \(g\) is by path \(abg(595 + 368 = 963)\) or path \(afg(419 + 541 = 960)\). The path \(afg\) yields the minimum time. The corresponding pressure path follows an average pressure of 144 for the 419 time period and then an average pressure of 134 for the 541 time period.

In a similar manner the minimum time from each node to the final

<table>
<thead>
<tr>
<th>Node</th>
<th>(x)</th>
<th>(y)</th>
<th>(z)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a)</td>
<td>0.60000</td>
<td>0.29765</td>
<td>0.10235</td>
</tr>
<tr>
<td>(b)</td>
<td>0.64000</td>
<td>0.26965</td>
<td>0.09035</td>
</tr>
<tr>
<td>(c)</td>
<td>0.68000</td>
<td>0.24165</td>
<td>0.07835</td>
</tr>
<tr>
<td>(d)</td>
<td>0.72000</td>
<td>0.21365</td>
<td>0.06635</td>
</tr>
<tr>
<td>(e)</td>
<td>0.76000</td>
<td>0.18565</td>
<td>0.05435</td>
</tr>
<tr>
<td>(f)</td>
<td>0.63000</td>
<td>0.27965</td>
<td>0.09035</td>
</tr>
<tr>
<td>(g)</td>
<td>0.67000</td>
<td>0.25165</td>
<td>0.07835</td>
</tr>
<tr>
<td>(h)</td>
<td>0.71000</td>
<td>0.22365</td>
<td>0.06635</td>
</tr>
<tr>
<td>(i)</td>
<td>0.75000</td>
<td>0.19565</td>
<td>0.05435</td>
</tr>
<tr>
<td>(j)</td>
<td>0.79000</td>
<td>0.16765</td>
<td>0.04235</td>
</tr>
<tr>
<td>(k)</td>
<td>0.66000</td>
<td>0.26165</td>
<td>0.07835</td>
</tr>
<tr>
<td>(l)</td>
<td>0.70000</td>
<td>0.23365</td>
<td>0.06635</td>
</tr>
<tr>
<td>(m)</td>
<td>0.74000</td>
<td>0.20565</td>
<td>0.05435</td>
</tr>
<tr>
<td>(n)</td>
<td>0.78000</td>
<td>0.17765</td>
<td>0.04235</td>
</tr>
<tr>
<td>(o)</td>
<td>0.82000</td>
<td>0.14965</td>
<td>0.03035</td>
</tr>
<tr>
<td>(p)</td>
<td>0.69000</td>
<td>0.24365</td>
<td>0.06635</td>
</tr>
<tr>
<td>(q)</td>
<td>0.73000</td>
<td>0.21565</td>
<td>0.05435</td>
</tr>
<tr>
<td>(r)</td>
<td>0.77000</td>
<td>0.18765</td>
<td>0.04235</td>
</tr>
<tr>
<td>(s)</td>
<td>0.81000</td>
<td>0.15965</td>
<td>0.03035</td>
</tr>
<tr>
<td>(t)</td>
<td>0.85000</td>
<td>0.13165</td>
<td>0.01835</td>
</tr>
<tr>
<td>(u)</td>
<td>0.72000</td>
<td>0.22565</td>
<td>0.05435</td>
</tr>
<tr>
<td>(v)</td>
<td>0.76000</td>
<td>0.19765</td>
<td>0.04235</td>
</tr>
<tr>
<td>(w)</td>
<td>0.80000</td>
<td>0.16965</td>
<td>0.03035</td>
</tr>
<tr>
<td>(x)</td>
<td>0.84000</td>
<td>0.14165</td>
<td>0.01835</td>
</tr>
<tr>
<td>(y)</td>
<td>0.88000</td>
<td>0.11365</td>
<td>0.00635</td>
</tr>
</tbody>
</table>
composition may be determined. Table 2 lists the step-by-step calculation to determine the minimum time from each node to the final composition [using Eqs. (19) and (20)]. For the nodes along the edges of the grid abcde and afkpu, we observe there is only one choice of the minimum path. For all other nodes, a choice must be made between the path at $ds/dx = -0.3$ and $ds/dx = -0.4$.

The cumulative minimum time from each node to the final composition at node $a$, as given in Table 2, is plotted in Fig. 9.

In Tables 3A and 3B are listed the minimum time paths from final point node $a$ to initial point node $y$, and the average pressure existing from node to node. In contrast to Table 2, the cumulative time at each node, presented in Tables 3A and 3B, is measured from the initial point $y$. The minimum time path is not unique in this example because, at
<table>
<thead>
<tr>
<th>Node</th>
<th>Time from node $j$ to node $(j-1)$</th>
<th>Total time from node $j$ to final point</th>
<th>Alternative total time from node $(j-1)$ to final point</th>
<th>Minimum total time from node $j$ to final point</th>
<th>Selected path</th>
</tr>
</thead>
<tbody>
<tr>
<td>$b$</td>
<td>595</td>
<td>595</td>
<td>595</td>
<td>$b$ to $a$</td>
<td></td>
</tr>
<tr>
<td>$c$</td>
<td>523</td>
<td>1118</td>
<td>1118</td>
<td>$c$ to $b$</td>
<td></td>
</tr>
<tr>
<td>$d$</td>
<td>460</td>
<td>1578</td>
<td>1578</td>
<td>$d$ to $c$</td>
<td></td>
</tr>
<tr>
<td>$e$</td>
<td>404</td>
<td>1982</td>
<td>1982</td>
<td>$e$ to $d$</td>
<td></td>
</tr>
<tr>
<td>$f$</td>
<td>419</td>
<td>419</td>
<td>419</td>
<td>$f$ to $a$</td>
<td></td>
</tr>
<tr>
<td>$k$</td>
<td>380</td>
<td>799</td>
<td>799</td>
<td>$k$ to $f$</td>
<td></td>
</tr>
<tr>
<td>$p$</td>
<td>347</td>
<td>1146</td>
<td>1146</td>
<td>$p$ to $k$</td>
<td></td>
</tr>
<tr>
<td>$u$</td>
<td>321</td>
<td>1467</td>
<td>1467</td>
<td>$u$ to $p$</td>
<td></td>
</tr>
<tr>
<td>$g$</td>
<td>368</td>
<td>595</td>
<td>963</td>
<td>$g$ to $f$</td>
<td></td>
</tr>
<tr>
<td>$h$</td>
<td>541</td>
<td>419</td>
<td>960</td>
<td>$h$ to $g$</td>
<td></td>
</tr>
<tr>
<td>$i$</td>
<td>323</td>
<td>1118</td>
<td>1441</td>
<td>$i$ to $h$</td>
<td></td>
</tr>
<tr>
<td>$j$</td>
<td>476</td>
<td>960</td>
<td>1436</td>
<td>$j$ to $l$</td>
<td></td>
</tr>
<tr>
<td>$l$</td>
<td>285</td>
<td>1578</td>
<td>1863</td>
<td>$l$ to $k$</td>
<td></td>
</tr>
<tr>
<td>$m$</td>
<td>421</td>
<td>1436</td>
<td>1857</td>
<td>$m$ to $l$</td>
<td></td>
</tr>
<tr>
<td>$n$</td>
<td>251</td>
<td>1982</td>
<td>2233</td>
<td>$n$ to $m$</td>
<td></td>
</tr>
<tr>
<td>$o$</td>
<td>369</td>
<td>1857</td>
<td>2226</td>
<td>$o$ to $n$</td>
<td></td>
</tr>
<tr>
<td>$q$</td>
<td>335</td>
<td>960</td>
<td>1295</td>
<td>$q$ to $p$</td>
<td></td>
</tr>
<tr>
<td>$r$</td>
<td>297</td>
<td>1436</td>
<td>1733</td>
<td>$r$ to $q$</td>
<td></td>
</tr>
<tr>
<td>$s$</td>
<td>435</td>
<td>1291</td>
<td>1726</td>
<td>$s$ to $r$</td>
<td></td>
</tr>
<tr>
<td>$t$</td>
<td>208</td>
<td>2444</td>
<td>2444</td>
<td>$t$ to $s$</td>
<td></td>
</tr>
<tr>
<td>$u$</td>
<td>308</td>
<td>1291</td>
<td>1599</td>
<td>$u$ to $q$</td>
<td></td>
</tr>
<tr>
<td>$v$</td>
<td>451</td>
<td>1146</td>
<td>1597</td>
<td>$v$ to $r$</td>
<td></td>
</tr>
<tr>
<td>$w$</td>
<td>273</td>
<td>1726</td>
<td>1999</td>
<td>$w$ to $v$</td>
<td></td>
</tr>
<tr>
<td>$x$</td>
<td>398</td>
<td>1597</td>
<td>1995</td>
<td>$x$ to $w$</td>
<td></td>
</tr>
<tr>
<td>$y$</td>
<td>238</td>
<td>2109</td>
<td>2347</td>
<td>$y$ to $x$</td>
<td></td>
</tr>
</tbody>
</table>

*The computed times are given in units of $10^{-5}$. For example, the time period of 541 units corresponds to the calculated values of 0.00541.*
point \( w \), the cumulative minimum time from point \( w \) to the node \( a \) is the same whether path \( wr \) or \( wv \) is taken. This may be seen in the listings of Table 2 under node \( w \). The two minimum time paths are \( yxvwqpkfa \) and \( yxwrqpkfa \). The average pressure-time profile may be plotted from the entries in Tables 3A and 3B.

**TABLE 3A**

<table>
<thead>
<tr>
<th>Path:</th>
<th>( y )</th>
<th>( x )</th>
<th>( w )</th>
<th>( v )</th>
<th>( q )</th>
<th>( p )</th>
<th>( k )</th>
<th>( f )</th>
<th>( a )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average pressure:</td>
<td>276</td>
<td>230</td>
<td>196</td>
<td>208</td>
<td>157</td>
<td>168</td>
<td>155</td>
<td>144</td>
<td></td>
</tr>
<tr>
<td>Cumulative time from initial point:</td>
<td>276</td>
<td>594</td>
<td>958</td>
<td>1238</td>
<td>1689</td>
<td>2036</td>
<td>2416</td>
<td>2835</td>
<td></td>
</tr>
</tbody>
</table>

**TABLE 3B**

<table>
<thead>
<tr>
<th>Path:</th>
<th>( y )</th>
<th>( x )</th>
<th>( w )</th>
<th>( r )</th>
<th>( q )</th>
<th>( p )</th>
<th>( k )</th>
<th>( f )</th>
<th>( a )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average pressure:</td>
<td>276</td>
<td>230</td>
<td>239</td>
<td>178</td>
<td>157</td>
<td>168</td>
<td>155</td>
<td>144</td>
<td></td>
</tr>
<tr>
<td>Cumulative time from initial point:</td>
<td>276</td>
<td>594</td>
<td>840</td>
<td>1238</td>
<td>1689</td>
<td>2036</td>
<td>2416</td>
<td>2835</td>
<td></td>
</tr>
</tbody>
</table>

We note in Tables 3A and 3B what appears to be an inconsistency in the pressure-time function. The pressure declines, rises, and then declines with time. For example, in Table 3A the average pressure declines as 276, 230, 196, then increases to 208, decreases to 157, increases to 168, and then declines. This behavior is due to the coarse grid chosen here. If a finer grid had been employed, the pressure would have been averaged over a shorter range of composition and time, and would have resulted in a smoother pressure-composition-time function.

**Discussion**

The minimum time path which we find by the dynamic programming approach is only the minimum consistent with the number of points and the permitted paths. As the grid is made finer and finer, the minimum time path calculated by this method approaches in the limit the "true" minimum for the continuous case. Since the technique is ideally suited for machine computation, it is an easy matter to alter the grid fineness to determine the effect on the minimum time, and the pressure-composition-time path.
The use of a grid is only a computational aid. It is not really necessary to solve the problem. In fact, just scattering points on the $xz$ plane and specifying that the minimum time path must pass through them will suffice. The grid, however, provides an orderly and simple scheme to traverse the $xz$ plane. In setting up an array of points by a grid or by any other device, one must check that it is possible for the manipulated variables to yield all the compositions. For this example a check on the maximum and minimum pressures guaranteed that the chosen values for the $dz/dx$ lines were satisfactory in this respect.

The dynamic programming approach has an important advantage over the calculus of variations solution to this problem; namely, by dynamic programming, inequality constraints are handled very conveniently. If, for example, a particular composition or pressure range were not admissible, the dynamic programming solution would test for these constraints and would choose a path that did not violate these constraints. On the other hand, the solutions to the Euler-Lagrange differential equations of the calculus of variations would give the minimum time path, unmindful of the inequality constraints.

One important problem that cannot be avoided by dynamic programming, calculus of variations, or any other method for that matter, is the problem of dimensionality. We have shown here how to handle a problem where only one variable, pressure, is manipulated. While the extension to more manipulated variables is straightforward, with more manipulatable variables, the number of decisions possible at each stage increases rapidly. The problem of dimensionality may tax the computer memory and speed. The inclusion of constraints reduces the number of acceptable combinations of the manipulated variables and often helps to reduce the quantity of computation. In complex chemical reactions, the state of the system may have to be specified by a number of constituents as well as by the manipulated variables (see Chapter 5, Section 9).

In the solution presented here the physical and chemical properties were evaluated at the mean pressure between adjacent nodes. This approximation becomes more accurate, of course, as the grid becomes finer.

We employed here in the numerical example the finite difference approximation to the differential equations rather than the exact solutions given in Eqs. (5) and (6). Had the analytical solutions been used, it would have been still necessary to evaluate the $k_i$ terms at some average conditions. The advantage of using the finite difference approximations rather than the differential equations is the generality of the method. This approach would have worked even if the equations did not have a solution in closed form.
Another approach to this problem is to combine the methodology of dynamic programming with that of the calculus of variations. This method yields a set of initial value partial differential equations (discussed in Sections 14–23).

It is interesting to observe that this numerical example illustrates the well-recognized fact that a desired terminal composition may be attained by many pressure-time routes.

Computer Control of the Batch Process

The dynamic programming solution to this batch reactor control problem gives the minimum time and the pressure-composition-time function to guide the process from the initial composition to the final composition. Once the pressure-time function is developed by the computer, the control computer can then act as a timing switch and change the pressure according to the optimum time schedule. From time to time in the control of a real process, the composition and pressure of the reacting system will be sampled and this information fed to the computer. Armed with this new information, the computer can regenerate the optimal pressure-time function for these new conditions, and guide the process accordingly.

Batch Hydrogenation Process

Eckman, Lefkowitz, and their associates at Case Institute of Technology have published a number of papers and theses on the computer control of a batch hydrogenation process [35, 36, 38, 43]. Their classic control work employs the calculus of variations to determine the minimum time to pass from a known initial composition to a final composition. Treating the problem as the brachistochrone problem they have found the minimizing time and the pressure-composition-time function for systems similar to that in Eq. (1).

In order to develop a tractable model, Eckman and Lefkowitz assumed that the reactions were carried out isothermally and that pressure was the only manipulated variable. Certain assumed relationships among the reaction velocity constants simplified the solution. In addition, in their work, inequality constraints were not considered.

We believe that dynamic programming offers an attractive method for solving these “path” processes which may or may not be solvable by the calculus of variations. Dynamic programming offers the following advantages:

(1) It operates directly on the differential equations of the process and does not require an analytical solution in closed form.
(2) It can handle equality as well as inequality constraints. In fact, an abundance of constraints aids, rather than hinders finding the solution.

(3) It is ideally suited for computer solution.

(4) Through it, the absolute minimum time path solution may be found. In addition, the next best time path may also be found. In fact, all the time paths from the minimum to maximum may be found. It may be of considerable economic advantage to know these alternate solutions and the penalties associated with operating for nonoptimal time periods (see Chapter 5, Section 23-24).

(5) Higher dimensionality problems can be handled more simply than by the calculus of variations, in many cases.

(6) The equations used by the dynamic programming method are much simpler than the Euler-Lagrange equations of the calculus of variations.

26. Comparison of Methods of Solution of the Brachistochrone Problem

We have presented several different ways to look at the brachistochrone class of problems.

By using a continuous dynamic programming version of the problem, we arrived at the basic equation.

$$f(x, y) = \min_{y'} \left[ \sqrt{\frac{1 + (y')^2}{2gy}} \Delta x + f(x + \Delta x, y + y'\Delta x) \right]$$  \hspace{1cm} (1)

Here the minimization is over the choice of $y'$ at the point $x, y$.

In the batch reaction problem, we employed a discrete version of the problem where the element of time was chosen from the finite difference approximation of the differential equations rather than from the integrated form of the differential equation as given in Eq. (6) of Section 24. In the batch reaction problem, the derivative terms $dx/dt$ and $dz/dt$ were not permitted to vary freely as in the brachistochrone problem. Rather, only certain arbitrarily chosen (derivatives) directions were permitted. In particular, only two directions were permitted at each point. While in principle we might have provided a larger number of permissible directions at each composition node, from a practical computing point of view this complicates the finding of the solution. In the batch reaction problem, the optimization was executed by finding the pressure which drove the composition from one node to adjacent nodes.
in minimum time along either of the two permitted slopes. In one sense, the optimization is over pressure; in another sense, the optimization is over the slopes. If the grid is fine enough, the true minimum path through it is approximated by moving piecewise from node to node along either of the two permitted paths.

By far, the biggest difference between the continuous and discrete version discussed here is the introduction of the grid with the two fixed slopes for the derivatives.

27. Dynamic Programming and Calculus of Variations
---Summing Up

In Sections 4-11, we have discussed some of the difficulties associated with the use of the calculus of variations. Now that we have developed some aspects of the dynamic programming point of view, we are prepared to summarize the utility of dynamic programming in the face of these difficulties.

a. Dynamic Programming and Boundary Values

One of the principal advantages of dynamic programming applied to variational problems is that the basic nonlinear partial differential equation [see Eq. (19), Section 14; Eq. (9), Section 15; and Eq. (13), Section 16] is an initial value problem rather than a two-point boundary value problem. Initial value problems are inherently easier to solve because the trial and error aspects of matching the two-point boundary conditions are simply not encountered.

On the other hand, the numerical evaluation of these equations, as Eq. (1) in Section 19, involves two incremental elements δ and Δ, whose relative size must be carefully selected to avoid stability and convergence problems.

The utilization of a discrete formulation, such as given in Section 20 in place of the continuous formulation in Section 14, provides a convenient way to solve variational problems. In dealing with the basic equations and constraints, we avoid some numerical problems, since there is only one increment, Δ, to be concerned with. We also might add that the Principle of Optimality provides a computational algorithm in a very natural way in this case.

If the basic nonlinear partial differential equation, Eq. (19) of Section 14, for example, is converted to a partial differential equation of the type Eq. (28) in Section 14, then a solution may be obtained by the method
of characteristics. This method is advantageous since the partial differential equation is solved by setting up an equivalent set of ordinary differential equations. The solution of ordinary differential equations is generally much simpler to obtain than the solution of a partial differential equation.

b. Dynamic Programming and Constraints

A dynamic programming formulation often requires a numerical solution. For this reason, constraints, equality, or inequality create no special problems. In applying the dynamic programming algorithm, we merely test to see if a constraint is violated. If it is, we rule out that combination of values of the variables as inadmissible. Since we have at our disposal the opportunity to select other combinations of variables, we can, by search procedures, find an admissible set, and even better, an optimal combination of variables.

Even problems with isoperimetric constraints can be handled in a fairly routine way by dynamic programming. These matters are discussed in Chapter 5, Section 19, in conjunction with the Lagrangian multiplier.

c. Dynamic Programming and Linearity

The problem of linearity does not raise any special questions in dynamic programming. In linear problems, the optimum lies at either end of the linear form. In the search procedure we can very efficiently scan the extremities of the linear constraints. In fact, linear problems can be solved by “brute force” search techniques that do not distinguish as to whether the equations are linear or not. Many classes of linear programming problems may be set up as dynamic programming problems.

d. Dynamic Programming and Nonanalytical Forms

Since many dynamic programming problems are solved by numerical means, nonanalytical forms can be handled readily (see Chapter 6, Section 11).

REFERENCES

3. Beckwith, R. E. Analytical and computational aspects of dynamic programming


44. Miele, A., Optimum burning program as related to aerodynamic heating for a missile traversing the earth's atmosphere. Jet Propulsion 27, No. 12, 1231-1240 (1957).


PROBLEMS

1. Let

\[ f(x, y) = \min_{\{y\}} \int_{x_1}^{x_2} F(x, y, y') \, dx \]

Show that

\[ f(x, y) = \min_{y'} \left[ F(x, y, y') A + f(x, y) + \frac{\partial f}{\partial x} A + \frac{\partial f}{\partial y} y'A + \ldots \right] \]

where \( A \) is an approximation to \( dx \).

By total differentiation with respect to \( x \) and partial differentiation with respect to \( y \), deduce the Euler-Lagrange equation

\[ \frac{d}{dx} \frac{\partial F}{\partial y} - \frac{\partial F}{\partial y} = 0 \]

(Dreyfus, S. [34].)

2. For a minimum to exist, the derivative of a function must be equal to zero and the second derivative must be greater than zero. Show that \( F_{y'y'} > 0 \). This is Legendre condition of the calculus of variations.

(Dreyfus, S. [34].)

3. To assure that an absolute minimum occurs rather than a relative minimum, show for all other derivatives \( Y' \) that the inequality

\[ F(x, y, y') + \frac{\partial f}{\partial y} + y' \frac{\partial f}{\partial y} \leq F(x, y, Y') + \frac{\partial f}{\partial x} + Y' \frac{\partial f}{\partial y} \]

holds which leads to the Weierstrass condition

\[ F(x, y, Y') - F(x, y, y') - (Y' - y') F_{y'} \geq 0 \]

4. Given

\[ f(x, y_1, y_2, \ldots, y_n) = \min_{\{y\}} \int_{x_1}^{x_2} F(x, y_1, y_2, \ldots, y_n, y_1', y_2', \ldots, y_n') \, dx \]
Show that

\[ 0 = \min_{y_1, y_2, \ldots, y_n} \left[ F + \frac{\partial f}{\partial x} + \sum_{j=1}^{n} y_j \frac{\partial f}{\partial y_j} \right] \]

Derive the set of Euler-Lagrange equations

\[ \frac{d}{dx} \frac{\partial F}{\partial y_i'} - \frac{\partial F}{\partial y_i} = 0, \quad i = 1, 2, \ldots, n \]

5. Suppose we desire to minimize \( \int_{x_0}^{x_1} F(x, y, y') \, dx \) subject to the familiar isoperimetric constraint.

\[ \int_{x_0}^{x_1} G(x, y, y') \, dx = Z = \text{constant} \]

Let us define

\[ f(x, y, Z) = \min_{\{y\}} \int_{x_0}^{x_1} F(x, y, y') \, dx \]

subject to the isoperimetric constraint.

Show that

\[ 0 = F_{y'} + \frac{\partial f}{\partial y} - G_{y'} \frac{\partial f}{\partial Z} \]  
(1)

\[ 0 = F + \frac{\partial f}{\partial x} + y' \frac{\partial f}{\partial y} - G \frac{\partial f}{\partial Z} \]  
(2)

From these equations develop

\[ \frac{d}{dx} \frac{\partial}{\partial y'} \left( F - \frac{\partial f}{\partial Z} G \right) - \frac{\partial}{\partial y} \left( F - \frac{\partial f}{\partial Z} G \right) = 0 \]  
(3)

By differentiating (2) with respect to \( Z \), show that \( \partial f/\partial Z = \text{constant} \).

Observe that \( \partial f/\partial Z \) plays the role of the Lagrangian multiplier, that the Lagrangian multiplier is independent of the independent variable \( x \), and that Eq. (3) is the Euler-Lagrange equation for an isoperimetric constrained problem.

6. Given

\[ \frac{dx_1}{dt} = a_1 y_1, \quad x_1(0) = c_1 \]

\[ \frac{dx_2}{dt} = (a_2 - 1) y_2 - y_1, \quad x_2(0) = c_2 \]

\[ f(x_1, x_2, T) = \max \{x_2(T)\} \]

\[ y_1, y_2 > 0, \quad y_1 + y_2 \leq c_2, \quad y_2 \leq \frac{b_1 c_1}{a_2}, \quad b_1 = \text{constant} \]
PROBLEMS

7. Given

\[ J(y) = \int_0^T F(x, y) \, dt \]

subject to

(a) \[ \frac{dx}{dt} = G(x, y), \quad x(0) = c \]

(b) \[ 0 \leq y \leq x \]

(c) \[ \int_a^T y \, dt \leq m \]

If we define

\[ f(a, c, m) = \max_y J(y) \]

show that

\[ f_T = \max_{0 \leq v \leq c} \left[ F(c, v) + G(c, v) f_e - vf_m \right] \]

(Bellman, R. [9].)

8. Minimize

\[ J(u) = \int_0^T [(u')^2 + g(u)] \, dt \]

with \( u(0) = c \). If we set

\[ f(c, T) = \min_u J(u) \]

show that

(a) \[ f_T = \min_v [v^2 + g(c) + vf_e] \]

where \( v = du/dt \) at \( t = 0 \)

(b) \[ v = -\frac{f_e}{2} \]

(c) \[ f_T = g(c) - \frac{f_e^2}{4} \]

(d) \[ f(c, 0) = 0 \] for all \( c \)

(Bellman, R. [24].)
9. Minimize the function

$$J(f) = \max_{0 \leq t \leq T} |x(t)|$$

subject to

$$x'' + \mu(x^2 - 1)x' + x - f(t) = 0$$

$$x(0) = c_1, \quad x'(0) = c_2, \quad \text{and} \quad |f| \leq 1$$

Show that the functional equations are given by

$$F_1(c_1, c_2) = |c_1|$$

$$F_{N+1}(c_1, c_2) = \max \left[ c_1, \min_{\mu_0 \leq 1} \left[ F_N(c_1 + \mu_0 c_2, c_2 + \Delta \{ f_0 - \mu(c_1^2 - 1)c_2 - c_1 \} \right] \right]$$

(Bellman, R. [13].)

10. Minimize $J(f)$ over $f(t)$, where

$$J(f) = \int_0^T [x^2 + (x'')^2 + af^2(t)] \, dt$$

subject to

$$x'' + \mu(x^2 - 1)x' + x = f(t); \quad x(0) = c_1, \quad x'(0) = c_2$$

Let

$$F_N(c_1, c_2) = \min_{f_{k_0}} J_N(f_k)$$

Show that

$$F_{N+1}(c_1, c_2) = \min_{f_{k_0}} [c_1^2 + \left( f_{k_0}^2 - \mu(c_1^2 - 1)c_2 - c_1 \right)^2 + af_{k_0}^2]$$

$$+ \left[ F_N(c_1 + \mu_0 c_2, c_2 + \Delta \{ f_0 - \mu(c_1^2 - 1)c_2 - c_1 \} \right]$$

$$F_1(c_1, c_2) = \min_{f_{k_0}} [c_1^2 + \left( f_{k_0}^2 - \mu(c_1^2 - 1)c_2 - c_1 \right)^2 + af_{k_0}^2]$$

11. If we are asked to minimize $J(f)$ over $f(t)$, where

$$J(f) = \max \int_0^T [x'' + \mu(x^2 - 1)x' + x - f(t)] \, dt$$

subject to

$$x^2 + (x')^2 + af^2(t) = 0; \quad x(0) = c_1, \quad x'(0) = c_2$$

why can it not be solved in this form? Is the following an acceptable formulation?

$$J(f) = \min \int_0^T [x^2 + (x')^2 + af^2(t)] \, dt$$
subject to

\[ x'' + \mu(x^2 - 1) x' + x = f(t); \quad x(0) = c_1, \quad x'(0) = c_2 \]

12. Suppose we desire to minimize

\[ J(u) = \int_0^T g(u, u') \, dt \]

subject to \( u(0) = c_1 \) and \( u(T) = c_2 \). If we let \( v = u' \) and \( u(t) = c \) and consider \( N\Delta = T \), show that the functional equations are

\[ f_{N+1}(c) = \text{Min}_v \left[ g(c, v) + f_N(c + v\Delta) \right] \]

\[ f_1(c) = g(c, v) = g\left(c, \frac{c_2 - c}{\Delta}\right) \]

Why is there no optimization in the one-stage process?

13. Given

\[ \frac{dx}{dt} = G_1(x, y, s_2) = r_1 - a_2 x - x\left[1 - \exp\left(-b_s y/x\right)\right] \]

\[ \frac{dy}{dt} = G_2(x, y, s_1) = r_2 - a_1 y - y\left[1 - \exp\left(-b_s y/y\right)\right] \]

where \( a_1, a_2, b_1, b_2, r_1, r_2 \) and \( s_2 \) are constants, and

\[ J(s_1) = \int_0^T ((1 - s_1) x - (1 - s_2) y) \, dt \]

Let \( x(0) = q_1 \) and \( y(0) = q_2 \) and define

\[ f(q_1, q_2, T) = \text{Max}_{s_1} J(s_1) \]

Show that for \( 0 \leq s_1(t) \leq 1 \)

\[ \frac{\partial f}{\partial T} = \text{Max}_{0 \leq s_1 \leq 1} \left[ (1 - s_1) q_1 - (1 - s_2) q_2 + G_1(q_1, q_2, s_2) \frac{\partial f}{\partial q_1} + G_2(q_1, q_2, s_1) \frac{\partial f}{\partial q_2} \right] \]

with \( f(q_1, q_2, 0) = 0 \).

(Bellman, R., and Dreyfus, S. [27].)

14. Given

\[ x_{k+1} = x_k + G_1[(x_k, y_k, s_2(k))] \Delta, \quad x_0 = q_1 \]

\[ y_{k+1} = y_k + G_2[x_k, y_k, s_1(k)] \Delta, \quad y_0 = q_2 \]
where
\[ x_k = x(k\Delta), \quad y_k = y(k\Delta), \quad s_{1k} = s_1(k\Delta), \quad s_{2k} = s_2(k\Delta) = \text{constant} \]

\[ J(s_1) = \sum_{k=0}^{N-1} (1 - s_{1k}) x_k - (1 - s_{2k}) y_k \]

with \(0 \leq s_{1k} \leq 1\). Let
\[ f_N(q_1, q_2) = \max J(s_1) \]

with
\[ f_1(q_1, q_2) = q_1 - (1 - s_{2k}) q_2 \]

Show that
\[ f_N(q_1, q_2) = \max_{0 \leq s_1 \leq 1} [(1 - s_1) q_1 - (1 - s_2) q_2 + f_{N-1}(q_1 + G_1(q_1, q_2, s_2) \Delta, q_2 + G_2(q_1, q_2, s) \Delta)] \]

\[ N = 1, 2, \ldots \]

15. Given
\[ x_{k+1} = x_k + G_1(x_k, y_k, s_{2k}, k), \quad x_R = q_1 \]
\[ y_{k+1} = y_k + G_2(x_k, y_k, s_{1k}, k), \quad y_R = q_2 \]

\[ f_R(q_1, q_2) = \max J_R(s_1) = \max \sum_{k=R}^{N} P(x_k, y_k, s_{1k}, k) \]

Show that
\[ f_R(q_1, q_2) = \max_{0 \leq s_1 \leq 1} [P(q_1, q_2, s_{1R}, R) + f_{R+1}(q_1 + G_1(q_1, q_2, s_2, R), q_2 + G_2(q_1, q_2, s_1, R))] \]

\[ R = 0, 1, 2, \ldots, N - 1 \]

with
\[ f_N(q_1, q_2) = \max_{0 \leq s_1 \leq 1} P(q_1, s_2, s_1, N) \]

16. Minimize \(\int_0^T x^2y \, dt\) where
\[ \frac{dx}{dt} = a_1 x + a_2 y + f, \quad x(0) = c_1 \]
\[ \frac{dy}{dt} = b_1 x + b_2 y + g, \quad y(0) = c_2 \]
\[ 0 \leq f(t) \leq R, \quad 0 \leq g(t) \leq Q \]
If we define $F_N(c_1, c_2) = \min_{f_k, g_k} \sum_{k=0}^{N-1} x_k^2 y_k$ and if we take

\begin{align*}
x_{k+1} &= x_k + \Delta(a_1 x_k + a_2 y_k + f_k) \\
y_{k+1} &= y_k + \Delta(b_1 x_k + b_2 y_k + g_k)
\end{align*}

show that

$$F_N(c_1, c_2) = \min_{f_0, g_0} \left[ c_1^2 c_2 + F_{N-1}(c_1 + \Delta(a_1 c_1 + a_2 c_2 + f_0), c_2 + \Delta(b_1 c_1 + b_2 c_2 + g_0)) \right]$$

(Bellman, R. [18].)

17. Minimize $\int_0^T x^2 y dt$ subject to

\begin{align*}
\frac{dx}{dt} &= a_1 x + a_2 y + f, \quad x(0) = c_1 \\
\frac{dy}{dt} &= b_1 x + b_2 y + g, \quad y(0) = c_2
\end{align*}

\[ \int_0^T f dt \leq M, \quad 0 \leq f \leq R \]

\[ \int_0^T g dt \leq m, \quad 0 \leq g \leq Q \]

Define

$$J_N(f_k, g_k) = \sum_{k=0}^{N-1} x_k^2 y_k + \lambda_1 f_k + \lambda_2 g_k$$

and

$$f_N(c_1, c_2) = \min_{f_k, g_k} J_N(f_k, g_k)$$

Show that

$$f_1(c_1, c_2) = \min_{f_0, g_0} \left[ c_1^2 c_2 + \lambda_1 f_0 + \lambda_2 g_0 \right]$$

$$f_N(c_1, c_2) = \min_{f_0, g_0} \left[ c_1^2 c_2 + \lambda_1 f_0 + \lambda_2 g_0 + f_{N-1}(c_1 + \Delta(a_1 c_1 + a_2 c_2 + f_0), c_2 + \Delta(b_1 c_1 + b_2 c_2 + g_0)) \right]$$

18. Minimize

$$J(f) = \int_0^T |1 - u| dt$$
where
\[ \frac{d^2u}{dt^2} + u = f(t); \quad u(0) = c_1, \quad u'(0) = c_2, \quad -1 \leq f(t) \leq 1 \]

Let
\[ \frac{du}{dt} = v, \quad u(0) = c_1 \]
\[ \frac{dv}{dt} = -u + f, \quad v(0) = c_2 \]
\[ u_{k+1} = u_k + \Delta v_k, \quad u_0 = c_1 \]
\[ v_{k+1} = v_k + \Delta (f_k - u_k), \quad v_0 = c_2 \]
\[ J_N(f_k) = \sum_{k=0}^{N-1} |1 - u_k| \]

Take
\[ F_N(c_1, c_2) = \text{Min } J_N(f_k) \]

Show that
\[ F_0(c_1, c_2) = |1 - c_1| \]
\[ F_{N+1}(c_1, c_2) = \text{Min}_{-1 \leq f_0 \leq 1} \left[ |1 - c_1| + F_N(c_1 + \Delta c_2, c_2 + \Delta (f_0 - c_1)) \right] \]

19. Minimize
\[ J(f) = \int_0^T |1 - u^2| \, dt \]

subject to
\[ \frac{d^3u}{dt^3} + \sqrt{u} = f(t) \]

Take
\[ u_{k+1} = u_k + \Delta v_k, \quad u_0 = c_1 \]
\[ v_{k+1} = v_k + \Delta w_k, \quad v_0 = c_2 \]
\[ w_{k+1} = w_k + \Delta [f_k - \sqrt{u_k}], \quad w_0 = c_3 \]

and define
\[ F_N(c_1, c_2, c_3) = \text{Min } J_N(f_k) = \text{Min } \sum_{k=0}^{N-1} |1 - u_k^2| \]

Show that
\[ F_0(c_1, c_2, c_3) = |1 - c_1^2| \]
\[ F_{N+1}(c_1, c_2, c_3) = \text{Min}_{-1 \leq f_0 \leq 1} \left[ |1 - c_1^2|, F_N(c_1 + \Delta c_2, c_2 + \Delta c_3, c_3 + \Delta (f_0 - \sqrt{c_1})) \right] \]
20. Given

\[ J(u) = \int_a^T u^2(x) \, dx + 2 \int_a^T u(x) \, v(x) \, dx + \int_a^T \int_a^T K(x, y) \, u(x) \, u(y) \, dx \, dy \]

where \( K(x, y) \) is a symmetric kernel over \( 0 \leq x, y \leq T \) and \( a \leq x \leq T \). The minimum of \( J(u) \) over all \( u(x) \) continuous in \( a \leq x \leq T \) is found from the solution of

\[ u(x) + v(x) + \int_a^T K(x, y) \, u(y) \, dy = 0 \]

For small \( s \) in the region \( a < a + s < T \), show that

\[
\begin{align*}
J(u) &= s[u^2(a) + 2u(a) \, v(a)] + 2 \int_{a+s}^T u(x) [v(x) + su(a) \, K(a, x)] \, dx \\
&= \int_{a+s}^T u^2(x) \, dx + \int_{a+s}^T \int_{a+s}^T K(x, y) \, u(x) \, u(y) \, dx \, dy + o(s)
\end{align*}
\]

By using the Principle of Optimality, develop

\[
f(v, a) = \min_u J(u) = \min_{u(a)} \left[ s[u^2(a) + 2u(a) \, v(a)] \right] \\
+ f(u(x) + su(a) \, K(a, x), a + s) + o(s)
\]


21. Given \( \int_a^T (u')^2 \, dt \) subject to the constraints

\[
\begin{align*}
& \int_a^T \phi(t) \, u^2 \, dt = 1 \\
& u(a) = k, \quad u(T) = 0
\end{align*}
\]

If we take

\[ f(a, k) = \min_u \int_a^T (u')^2 \, dt \]

show that

\[ f_T = - \frac{f_k^2}{4} + \frac{k^2\phi(a)f_k}{2} - k^2\phi(a)f \]

(Bellman, R. [9].)
1. Introduction

This chapter deals with some computational aspects of dynamic programming. As with most analytic methods, dynamic programming rarely yields exact analytical results. Consequently, computational aspects of this discipline are extremely important. As befitting a modern mathematical method, dynamic programming requires modern mathematical devices, namely, high-speed computing machinery. The marriage of a powerful mathematical approach and powerful computing machinery has opened new frontiers and made possible the solution of many problems formerly beyond our grasp.

In Chapter 5, we attempt to give an over-all point of view to the computational aspects of dynamic programming. We point out the standard way of solving problems and also the conditions that arise to frustrate the normal approach. Chief among these is the problem of high dimensionality whose resolution is discussed in a number of sections. At various places in the text, detailed numerical problems are worked out. Since these problems are simple enough to be solved without computers, the reader should find it worthwhile to develop by hand the results for himself.

Let us now briefly review the chapter. We discuss in Section 2 the need for computation in dynamic programming. Section 3 highlights the difference between the combinatorial approach and the dynamic
programming approach with a simple numerical example employing both techniques. In Sections 4–9, a discussion of computational techniques for discrete problems is given. We also point out the problem of high dimensionality and how it may be coped with to some extent. A comparison of the dynamic programming approach and the differential calculus approach to the allocation problem is covered in Section 10. The next few sections are devoted to the subjects of successive approximations, approximations in function space, and approximations in policy space. A simple allocation problem is solved by several different methods to show that in the limit, all the methods yield the same results. A discussion of analytical results for certain special return functions for the allocation problem is given in Section 18. The very important topic of Lagrangian multipliers with an example of how it normally works and a counterexample of where it works only partially also is covered. Following this, we discuss the kth best policy and illustrate in detail an example. Sections 25–30 deal with techniques to reduce dimensionality such as mechanical quadrature, linearization, and successive linear approximation. We point out in Section 31 the difficulties associated with problems whose state is described by a distribution function, rather than by discrete variables. Finally, we conclude this chapter with a discussion of the dynamic programming approach to problems normally considered in the realm of linear programming.

2. Computational Aspects, Raison d'être

Dynamic programming problems are often solved by numerical means, using computers, rather than by exact analytical solutions. In this lies both the strength and weakness of dynamic programming.

Numerical solutions are an integral part of dynamic programming because:

1. Some problems not solvable analytically by other techniques can be formulated by dynamic programming.
2. The dynamic programming formulation does not generally yield exact analytical results.
3. Some problems are constrained by inequality constraints, algebraic or isoperimetric.
4. Some problems may contain unusual terms such as the absolute value of function.
5. Some problems may contain discontinuous functions.
6. Some problems may be combinatorial of high dimension.
Dynamic programming has become a refuge for problems unsolvable by other techniques, especially exact analytical techniques. Casting the problem into a dynamic programming guise may resolve problems that have previously defied solution. The consideration, for example, of many calculus of variations problems, as multi-stage processes renders a solution not attainable by conventional techniques.

The formulation of a problem as a dynamic programming problem is only half the story. It is necessary to solve the functional equations. In general, this cannot be done analytically and recourse must be had to numerical methods. At our disposal, however, is the vast and growing field of numerical analysis. Numerical techniques in conjunction with high-speed digital computers take the problem formulated by dynamic programming and generate numerical answers. While numerical approaches to problems have been criticized in the past as lacking the generality of the analytical solutions, this criticism cannot be leveled at dynamic programming. By virtue of the imbedding process, the numerical results cover a breadth of conditions to effectively give the same results as an analytical solution.

Before we dismiss the analytical aspects of dynamic programming too lightly, let us mention that there are problems that do lend themselves to exact analytical results. Such results when found, are extremely useful since they can be used to generate the optimal policy without computation. In addition, analytical results of simple problems can be used as approximations for more complicated problems. In Section 18, the analytical results for certain allocation problems are given.

A common method for solving dynamic programming problems is to set up a grid in the variables. Each node of the grid represents a set of numerical values for the variables. By search techniques, the various nodes of the grid are explored to find the optimum node. This technique has the advantage of being readily programmed for a computer but it has the disadvantage of requiring a great deal of computer time and memory when the grid is fine and/or the number of variables is larger than, say, four.

The numerical approach makes it quite easy to test for inequality constraints. If the selection of a node on the grid causes a constraint to be violated, that node is rejected. Adjacent nodes are tested until it is possible to delineate the permissible region on the grid. The larger the number of constraints the smaller is the permissible region with no particular hardship computationally. In contrast to this the resolution of problems by exact analytical methods becomes more and more involved as the number of constraints increases.

Numerical methods thankfully are oblivious of the difficulties associat-
ed with handling unusual functions, such as the absolute value of a function. To a computer the generation of a positive, negative, or absolute valued function is a matter of small concern.

Similarly, discontinuous functions are handled readily by numerical means. Price or cost structures are often discontinuous functions of the cumulative volume of a product. The numerical evaluation of the cost can be done by a table-look-up or by equations fitted over the separate parts of the discontinuous function.

One of the strengths of the dynamic programming point of view is that it effectively reduces the quantity of computation. Instead of trying to solve the entire problem at one fell swoop, it solves the problem step by step. For example, in an $N$-stage process where there are $k$ possible decisions, the combinatorial approach requires listing and ranking at one time the entire $k^N$ possibilities. Dynamic programming, on the other hand, at each stage examines the $k$ possibilities. As a result, only $Nk$ possible decisions over the $N$ stages must be examined. In other words, dynamic programming yields the same results as the combinatorial method for $Nk/k^N = N/k^{N-1}$ the effort. The significance of this ratio is the difference between solving and not solving a problem for large $k$ and/or $N$. The time to examine $k^N$ possibilities may be prohibitive even for the largest computers.

The dimensionality of the problem as given by the number of manipulated and state variables often require high dimensional grids. This is a real drawback. With increasing dimensionality of the grid as well as with increasing fineness of mesh size, the computer space and speed become limiting. A great deal of effort and attention has been paid to high-dimensional problems. If we had to choose today the one principal limitation to the application of dynamic programming, without doubt, we would select the problem of high dimensionality. In this chapter, we will explore some devices and dodges to reduce dimensionality.


To understand better the advantages of dynamic programming over the combinatorial approach, let us consider a simple critical path scheduling problem in Fig. 1. It is desired to find the least time path from node 1 to node 6.

This is a directed network where one must proceed only in the direction of higher nodal numbers. For example, the paths 3 to 4 and 3 to 5 are permitted but the paths 3 to 2 or 3 to 1 are not.
The time of travel from node to node is given on the line segment joining the nodes.

**FIG. 1.**

**Combinatorial Approach**

The combinatorial approach considers all possible paths from node 1 to node 6 and selects the least time path. All the possible paths and corresponding times are listed and ranked in the accompanying tabulation. The least time path is 1, 2, 5, 6 and the corresponding time is 7 units.

<table>
<thead>
<tr>
<th>Rank</th>
<th>Path</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1 2 5 6</td>
<td>7</td>
</tr>
<tr>
<td>2a</td>
<td>1 3 5 6</td>
<td>8</td>
</tr>
<tr>
<td>2b</td>
<td>1 2 3 5 6</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>1 2 4 6</td>
<td>13</td>
</tr>
<tr>
<td>4a</td>
<td>1 2 3 4 5 6</td>
<td>15</td>
</tr>
<tr>
<td>4b</td>
<td>1 2 3 4 6</td>
<td>15</td>
</tr>
<tr>
<td>4c</td>
<td>1 3 4 5 6</td>
<td>15</td>
</tr>
<tr>
<td>4d</td>
<td>1 3 4 6</td>
<td>15</td>
</tr>
</tbody>
</table>

**Dynamic Programming Approach**

The dynamic programming approach in contrast to the combinatorial approach considers the least time from each node to the final node. By a bootstrap operation, one works backward from the final node 6 to the
initial node 1. For example, starting at node 4, one may reach 6 by the path 4 to 6 or by the path 4, 5, 6. Either way, the time is identical, namely, 5 units. The knowledge of the least time from 4 to 6 is then used to compute the least time from nodes 2 and 3, without going through the combinatorial process. A detailed numerical account is given below.

Let

\[ t_{ij} \] is the time to travel from node i to node j

\[ f_i = \text{the minimum time to pass to node } i \text{ to the final node, node 6, along the permitted paths and using an optimal policy} \]

The functional equations are

\[ f_i = \text{Min} \left( t_{ij} + f_j \right) \]

\[ f_6 = 0 \]

\[ f_5 = \text{Min} \left( t_{5,6} + f_6 \right) = \text{Min} \left( 2 + 0 \right) = 2 \]

\[ f_4 = \text{Min} \left( t_{4,5} + f_5 \right) = \text{Min} \left( 3 + 2 = 5 \right) = 5 \]

\[ f_3 = \text{Min} \left( t_{3,4} + f_4 \right) = \text{Min} \left( 7 + 5 = 12 \right) = 5 \]
4. General Computation Process

We define

\[ f_N(x) = \text{the maximum return over the } N \text{ remaining stages of a process} \]

beginning in state \( x \), and using an optimal policy. Here \( N = N \) refers to
the first stage, \( N = 1 \) refers to the last stage.

Let us consider the functional equations:

\[ f_N(x) = \max_{0 \leq y_N \leq x} \left[ g(y_N) + f_{N-1}(x - y_N) \right] \] (1)

\[ f_1(x) = \max_{0 \leq y_1 \leq x} [g(y_1)] = g(x) \] (2)

This problem is a typical allocation problem, where the resource \( x \) is to be shared over \( N \) stages of time. The maximization is executed by the proper choice of \( y_N \) in stage \( N \) over the range \((0, x)\). During stage \( N \), for the proper choice of the resource \( y_N \), the process earns a return of \( g(y_N) \) and over the \((N - 1)\) remaining stages, the process earns a cumulative return \( f_{N-1}(x - y_N) \).

The solution to this typical equation is based on a recursion relationship between the return for an \( N \)-stage process and a \((N - 1)\)-stage process, \( f_N(x) \) and \( f_{N-1}(x - y_N) \), and between the transformation from state \( x \) to state \((x - y_N)\). The stages are counted “backward”\( ^{\dagger} \) so

\[ f_N(x) = \max_{0 \leq y_1 \leq x} [g(y_1) + f_{N-1}(x - y_1)] \]

\[ f_1(x) = \max_{0 \leq y_1 \leq x} g(y_1) = g(x) \]

\( \dagger \) We need not count the stages backwards always. In some problems, we count forward.

If we count the stages forward so that \( N = 1 \) refers to the first stage and \( N = N \) refers to the last stage, the counterparts to Eqs. (1) and (2) are, respectively:

\[ f_N(x) = \max_{0 \leq y_1 \leq x} [g(y_1) + f_{N-1}(x - y_1)] \]

\[ f_1(x) = \max_{0 \leq y_1 \leq x} g(y_1) = g(x) \]

The cumulative times and directed paths are shown in Fig. 2.

If we compare the least time path found by dynamic programming with that found by the combinatorial method, we see that we have identical results.

While we have illustrated here only a simple problem, the advantage of dynamic programming becomes more and more prominent as the combinatorial feature becomes magnified.
that $N = 1$ means one stage remaining and $N = N$ means the beginning of the process with $N$ stages remaining. Starting with the last stage, the term $f_1(x)$ is found first for a number of $x$ values. The term $f_3(x)$ is found from Eq. (1) by choosing the proper $y_3$ for stage 2, so that when the return $g(y_2)$ is added to the return over the one remaining stage $f_1(x - y_3)$, the sum yields a maximum over the total of two stages. In this bootstrap manner, the function $f_N(x)$ can be evaluated for any $N$ and any $x$.

It is convenient to select discrete values for $x$, namely, $x_1$, $x_2$, $x_3$, ... and to set up a table as shown in Table 1. The evaluation of Eq. (2) gives the first line of the Table 1. The evaluation of Eq. (1) gives all the other lines in the table successively. It is important to note that, say, for stage 3 and the determination of $f_3(x_2)$, that the choice of $y_3$, which is generally a trial and error search process, determines the state at the beginning stage 2. For a table of sufficiently closely spaced values of $x$, namely, $x_1$, $x_2$, $x_3$, etc., the choice of $y_3$ for stage 3 will cast the process into the new state $(x_2 - y_3) = x_3$, or $x_4$, or $x_5$, etc. For a coarser grid it may be necessary to interpolate in the table, since the choice $y_3$ may yield an $x_i$ which is not tabulated. It is also possible to consider only discrete increments in $y$ as well as discrete increments in $x$.

**TABLE 1**

<table>
<thead>
<tr>
<th>Number of stages remaining</th>
<th>State of the system</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$0$</td>
</tr>
<tr>
<td>1</td>
<td>$f_1(0)$</td>
</tr>
<tr>
<td>2</td>
<td>$f_1(0)$</td>
</tr>
<tr>
<td>3</td>
<td>$f_1(0)$</td>
</tr>
<tr>
<td>4</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
<tr>
<td>$N$</td>
<td>$f_N(0)$</td>
</tr>
</tbody>
</table>

From the solution to Eqs. (1) and (2) are determined (1) the over-all optimal return for the $N$ remaining stages, (2) the return for each stage, and (3) the policy of generating the optimal return. The policy in this case is the set of functions $y_i$ that maximizes the right-hand side of Eqs. (1) and (2); namely, $y_1^*, y_2^*, ..., y_N^*$. From Table 1, it is possible to trace the path from state to state and from stage to stage.
As a general rule, it is a good idea to start the computation using a coarse grid. This reduces the quantity of computation and gives a first approximation to the optimal process. Increasing the fineness of the mesh of the grid can be tried next to see the effect on the return structure and the policy structure. Proceeding from coarse grid to fine grid, one will reach the point where the fineness of the grid will not materially alter the return or policy function. No matter what grid spacing is used, it may be necessary to interpolate. This may come about due to the fact that over the range of $y$ values, the optimal value of $y$ generates an $(x - y)$ term, as the state for the next stage, which value is not tabulated in the Table 1, but lies between two values that are tabulated. In processes where more than one variable may be used to alter the state of the system, interpolation becomes more common.

The details of the computation process may be cleared up by referring to the numerical examples given in Chapter 2, Sections 3 and 9.

In the solution to dynamic programming problems, the novice is often disturbed by the questions:

1. How does one know the final state of the system?
2. How does one know how many stages to consider?

To answer the first question, we point out that one does not know in general the final state of the process. One does, however, know the possible ranges of states. To cover the range of final states, we consider discrete states $x_1, x_2, \ldots, x_k$, and evaluate $f_1(x_1), f_1(x_2), \ldots, f_1(x_k)$. In the generation of the solution, the final state will be determined as a result and need not be specified beforehand to embark on the calculation. Even in those problems where the final state is specified, the technique described above works. In addition, it gives information regarding the range of final states and the corresponding range of initial states.

To answer the second question, one arbitrarily sets the number of stages high enough so that the policy is independent of the number of stages. By that we mean the number $N$ is chosen large enough so that for all practical purposes, the $N$-stage process approximates an infinite stage process. To be more specific, if the catalyst replacement policy for a 40-stage, 30-stage, 20-stage, and 10-stage process is an 8-stage cycle, then the minimum number of stages to be considered is 8. To use a process with less than 8 stages will not yield a valid replacement policy. To use a process with more than 8 stages adds no additional information. For all practical purposes, an 8-stage process is equivalent to an infinite stage process. We are concerned in a 40-, 30-, 20-, or 10-stage process not with the profit over all 40, 30, 20, or 10 stages, but really with only the profit over the catalyst replacement period. The
only way to determine the catalyst replacement period is to solve the problem for a large enough number of stages and let the solution disclose it.

In some problems, the number of states is specified by the system being considered. If the allocation of feed is carried out over, say, 5 reactors, then we obviously have a 5-stage process.

In other problems, such as determining the optimal length of a tubular reactor or determining the number of stages of heat exchange, we leave the problem “open-ended” so to speak, and examine the relative return from the various stages. In these cases, as in the catalyst replacement example, the solution itself determines the optimal number of stages.

5. Expanding Grid Problem

In the problem discussed in Section 4, we have tacitly understood that the range of states of the system decreased as the number of stages decreased. There are, however, many problems where the range of states of the system increases as the stages decrease. A typical example of this is furnished by the equation

\[ f_N(x) = \max_{y_N} \left[ g(y_N) + f_{N-1}(x + \Delta x) \right] \]  \hspace{1cm} (1)

where

\[ \Delta x = h(y_N) \]  \hspace{1cm} (2)

The numerical evaluation of this involves generating larger and larger sets of grid points for the x terms. In other words, we have an expanding grid problem. In order to evaluate \( f_N(x) \) over a range of x’s for the Nth stage may require an extremely large range of x’s for stage 1.

In some cases, as discussed in Section 35 on linear programming, the expanding grid problem may be eliminated by suitable reformulation.

6. Two-Dimensional Problems

Problems in which there is only one state variable can be handled very neatly by the table as given in Table 1. If two state variables are required, it is possible to set up a grid in the two state variable x and \( z \) and compute the tableau with headings as in Table 2.

It is readily seen that a problem with two state variables can occupy
### Table 2

<table>
<thead>
<tr>
<th>Number of stages remaining</th>
<th>$x_1, z_1$</th>
<th>$x_1, z_2$</th>
<th>$x_1, z_3$</th>
<th>$x_2, z_1$</th>
<th>$x_2, z_2$</th>
<th>$x_2, z_3$</th>
<th>$x_3, z_1$</th>
<th>$x_3, z_2$</th>
<th>$x_3, z_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$f_1(x_1, z_1)$</td>
<td>$f_1(x_1, z_2)$</td>
<td>$f_1(x_1, z_3)$</td>
<td>$f_1(x_2, z_1)$</td>
<td>$f_1(x_2, z_2)$</td>
<td>$f_1(x_2, z_3)$</td>
<td>$f_1(x_3, z_1)$</td>
<td>$f_1(x_3, z_2)$</td>
<td>$f_1(x_3, z_3)$</td>
</tr>
<tr>
<td>2</td>
<td>$f_2(x_1, z_1)$</td>
<td>$f_2(x_1, z_2)$</td>
<td>$f_2(x_1, z_3)$</td>
<td>$f_2(x_2, z_1)$</td>
<td>$f_2(x_2, z_2)$</td>
<td>$f_2(x_2, z_3)$</td>
<td>$f_2(x_3, z_1)$</td>
<td>$f_2(x_3, z_2)$</td>
<td>$f_2(x_3, z_3)$</td>
</tr>
<tr>
<td>$N$</td>
<td>$f_N(x_1, z_1)$</td>
<td>$f_N(x_1, z_2)$</td>
<td>$f_N(x_1, z_3)$</td>
<td>$f_N(x_2, z_1)$</td>
<td>$f_N(x_2, z_2)$</td>
<td>$f_N(x_2, z_3)$</td>
<td>$f_N(x_3, z_1)$</td>
<td>$f_N(x_3, z_2)$</td>
<td>$f_N(x_3, z_3)$</td>
</tr>
</tbody>
</table>
considerably more space in a computer and can require considerably more time for solution than a single state variable problem.

While the state of a system may be specified by a single variable, there may be a number of manipulated variables which may affect the objective function, (the profit or cost) without altering the state. For example, in the catalyst replacement problem, Chapter 2, Sections 7 and 8, both temperature and flow rate may be manipulated to change the profit, yet only the instantaneous flow rate altered the state of the system, which was specified by the cumulative flow rate.

7. Multi-Dimensional Problems

Multi-dimensional variable problems at our present state of knowledge represent one great limitation on the application of dynamic programming. Dimensionality, as discussed here, arises in two areas: the first is in the description of the state of the system, the second is in the variables available for manipulation. It is often convenient to describe the state space by a grid in the state variables. Similarly, it is often convenient to permit the manipulated variables to assume only certain discrete values on a lattice. In some problems, the state variables and the manipulated variables are identical (see Chapter 3, Section 2). In other problems, the number of manipulated variables exceeds the number of state variables (see Chapter 2, Section 7). While still in other problems, the number of state variables exceeds the number of manipulated variables (see Chapter 4, Section 25). For high dimensionality in the state variables and in the manipulated variables, the standard approach leads to multi-dimensional grids. Problems of high dimensionality are usually attacked by side thrusts or rear assaults rather than by direct methods. The direct assault on high-dimensionality problems requires an extraordinary amount of computer space and time. As a practical upper limit, a problem in six dimensions is about as large a problem that can be considered on modern high-speed computing machinery. Certainly, one fruitful avenue for research in dynamic programming is the area of high dimensionality.

The dimensionality problem is side-stepped by a number of artifices outlined below and discussed in later sections. Each permits an answer to be generated of more or less satisfactory nature. It should be remembered here, that where dynamic programming is limited by dimensionality, so are other techniques.†

† Many deterministic problems of high dimension, however, can be formulated and solved by linear programming, quadratic programming, and gradient methods.
8. Step-by-Step Grid Generation

Problems of high dimensionality can be handled by the following:

1. Linearizing the problem and using linear programming.
2. Linearizing equations and using dynamic programming (see Sections 27–30 and 32–36).
3. Employing a Lagrangian multiplier. This is discussed in Sections 19–22.
4. Developing a variable grid system, rather than a fixed grid (see Sections 8 and 9).
5. Parameter studies where certain variables are held at fixed levels (see Section 9).
6. Restriction of the range over which variables from stage to stage may vary (see Section 9).
7. Predetermining how certain variables shall change from stage to stage (see Section 9).
8. Approximating by quadratic programming.
10. Polynomial approximations (see Section 26).

In our discussion so far, we have talked about setting up a grid in the state variables at discrete intervals (often evenly spaced), and computing the return function. The choice of the grid spacing has been set before attempting the solution of the problem. This implies that each node of the grid is a physically realizable point.

In many problems, it is not possible to lay out an \textit{a priori} grid that is meaningful, since we may not know for certain that the nodes are physically realizable.

To overcome this difficulty, we seek another approach. The approach we recommend is the generation of the grid step-by-step starting from an initially known physically real point. From the initial point, we step off certain increments in certain directions. We then test, using the equations of the process to see if any of these trial points are physically meaningful. If they are, we have established one or more real points, each of which can be used as a nucleus for stepping off again, with various increments in certain directions. If at the initial point after the first attempt we do not establish other physically real points around it, we continue trying by altering either the step size or the direction. In this manner, we establish the entire grid by showing that each node of the process represents true physical conditions. In the generation of the
grid step by step, we do not necessarily employ fixed increment sizes or fixed directions. The grid mesh is adapted to accommodate the needs of the process being described.

9. Example of Step-by-Step Grid Generation. Control of Batch Reaction

To illustrate the need for the step-by-step grid generation, we will refer to the example of the batch reaction control given in Chapter 4, Section 25. While the example originally was solved in terms of a two-dimensional grid, it might equally well have been solved by a three-dimensional grid. We will discuss first the problems involved in setting up a three-dimensional grid. Following this we pass on to methods for handling problems of higher dimensionality.

a. Review of Batch Reaction Control Example

In Chapter 4, Section 25 the following batch reaction was considered:

\[ X \xrightarrow{k_1} Y \xrightarrow{k_2} Z \]  

(1)

The equations describing the process were:

\[
\frac{dx}{dt} = -k_1 x + k_3 y, \quad \frac{dy}{dt} = +k_1 x - (k_2 + k_3)y, \quad \frac{dz}{dt} = +k_2 y
\]  

(2)

where \(x + y + z = 1\).

The coefficients \(k_i\) were taken to be functions of pressure alone. The purpose of control was to drive the process from a known composition \((x_0, y_0, z_0)\) to a desired final composition \((x_F, y_F, z_F)\), in a minimum of time by manipulation of the pressure. To carry out the optimization, a grid was set up on the \(xz\) plane so that the process was constrained to move along lines where the ratios

\[
\frac{\Delta z}{\Delta x} = -C_1 \quad \text{or} \quad \frac{\Delta z}{\Delta x} = -C_2
\]  

(3)

were constant.

The grid was chosen in terms of \(x\) and \(z\), since \(x\) and \(z\) were known in advance to be monotonic. In particular, \(x\) always decreased and \(z\) always increased with time. While a composition grid using \(y\) as a coordinate might have been employed, it was not, since \(y\) increases to a
maximum and then decreases. The \( yx \) grid, or the \( yz \) grid for this case, would have introduced the complication of changing grids after \( y \) peaked. This complication was avoided by the \( xz \) grid. It also was pointed out that it was not possible to use a grid of \( x \) and \( z \) with the elements of \( x \) and \( z \) parallel to the \( x \) and \( z \) axes, respectively. This implied that \( dx/dt, dy/dt, \) and \( dz/dt \) were all equal to zero, which are conditions for equilibrium and imply no change in the state of the process.

On each leg of the grid we marked both the time and the average pressure required to drive the process from node \( k \) to node \((k + 1)\). Using this information and the Principle of Optimality, the minimum time path can be found.

The equations are:

\[
f_N(x_N, z_N, P_N) = \min_{P_{\text{min}} \leq P \leq P_{\text{max}}} \left\{ \begin{array}{l} \Delta t_N + f_{N-1}[x_N + \Delta t_N(-k_1x_N + k_3y_N), \\
\delta t_N + f_{N-1}[x_N + \delta t_N(-k_1x_N + k_3y_N), \\
\delta t_N = \Delta t_N k_3 y_N, P_{N-1}] 
\end{array} \right. 
\]

(4)

\[
f_1(x_1, y_1, P_1) = \min_{P_{\text{min}} \leq P \leq P_{\text{max}}} [\Delta t_1; \delta t_1] 
\]

(5)

where

\[
f_N(x_N, z_N, P_N) = \text{the minimum time for the system to move from the current state } (x_N, z_N, P_N) \text{ over the } N \text{ remaining stages to the final composition } (x_F, z_F), \text{ using an optimal policy. The subscript } N \text{ refers to the number of stages remaining, that is to say, to the number of nodes in the grid which must be traversed to reach the final composition. When one node away from the final composition, } N = 1. N = 1, 2, ...
\]

\[
\Delta t_N = \text{the time to travel from node } N \text{ to node } (N - 1) \text{ along the path } dx/dx = -C_1
\]

\[
\delta t_N = \text{the time to travel from node } N \text{ to node } (N - 1) \text{ along the path } dy/dx = -C_1
\]

b. Case I

**State of the System Specified by \( x, z, v_1 \)**

For the system described by Eq. (1), where the coefficients \( k_i \) are functions only of one variable, the state of the system is specified by \( x, z, \) and \( v_1 \) (where the variable \( v_1 = \) the pressure in Chapter 4, Section 25). It appears reasonable to set up a three-dimensional grid in \( x, z, \)
and $v_1$, where the equations of the elements of the grid are described by:

$$\frac{Ax}{a_1} = \frac{Az}{a_2} = \frac{Av_1}{a_3}$$

$$\frac{Ax}{b_1} = \frac{Az}{b_2} = \frac{Av_1}{b_3}$$

$$\frac{Ax}{e_1} = \frac{Az}{e_2} = \frac{Av_1}{e_3}$$

The quantities $a_1, a_2, a_3, b_1, b_2, b_3, e_1, e_2,$ and $e_3,$ are the direction numbers. Since the direction numbers are known, fixing the size of one of the $Ax$, $Az$, or $Av_1$ terms determines the size of the other increments. The three-dimensional grid will then represent all combinations of $x$, $z$, and $v_1$ consistent with the grid spacing, and the permitted paths of travel along the legs of the grid (which are specified by the direction numbers). Once the three-dimensional grid has been selected in advance, it can be seen that solving Eqs. (2) would determine the time to pass from node $x_j$, $z_j$, $(v_1)_j$ to node $x_{j+1}$, $z_{j+1}$, $(v_1)_{j+1}$. Then Eqs. (4) and (5) can be used to solve for the minimum time path.

The only trouble with this three-dimensional representation of the $x$, $z$, $v_1$ system is that it will not work in general. The reason for this is that while it is possible to set up a grid of three-tuples beforehand, one does not know beforehand whether the three-tuples are physically realizable. If we start at $x_0$, $z_0$, $(v_1)_0$, along the path indicated by the direction numbers $a_1$, $a_2$, $a_3$, the equation

$$\frac{Ax}{Ax} = \frac{Az}{a_1} = \frac{k_2y}{-k_1x + k_3y}$$

must be satisfied. The variables $k_i$ are functions here of $v_1$ only and are evaluated at $\frac{1}{2} [(v_1)_0 + (v_1)_1]$. Unless a very fortuitous choice of the $v_1$ grid is made, Eq. (7) will not be satisfied. This is equivalent to saying the transition from state $[x_0$, $z_0$, $(v_1)_0]$ to $[x_1$, $z_1$, $(v_1)_1]$ is physically impossible. It is very likely that if the three-dimensional grid is set up arbitrarily, it will not be possible to move from any given node to any of its adjacent nodes.

To illustrate the incompatibility of fixing an arbitrary three-dimensional grid, let us examine Tables 3, 4, 5A, and 5B, which are based on the data in Chapter 4, Section 25. In Table 3, we specify direction numbers and the increment size. In column 4 of Table 4 is given the value of $Az/Ax$ corresponding to each direction. In columns 5, 6, and 7 of Table 4, we list the coordinates of the nodes adjacent to the initial point. We also give, in column 8, the average pressure between the
initial pressure and the pressure at each node (see Fig. 3). In Table 5A and 5B, Eq. (7) is evaluated for the values of \( \Delta z/\Delta x \) which are given in

### TABLE 3

<table>
<thead>
<tr>
<th>Direction number</th>
<th>( +\Delta x )</th>
<th>( \Delta z )</th>
<th>( \Delta P )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( a )</td>
<td>0.4</td>
<td>800</td>
<td></td>
</tr>
<tr>
<td>( b )</td>
<td>0.3</td>
<td>800</td>
<td></td>
</tr>
<tr>
<td>( e )</td>
<td>0.2</td>
<td>666.7</td>
<td></td>
</tr>
</tbody>
</table>

Increment size

<table>
<thead>
<tr>
<th>Direction number</th>
<th>( +\Delta x )</th>
<th>( \Delta z )</th>
<th>( \Delta P )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( a )</td>
<td>0.012</td>
<td>24</td>
<td></td>
</tr>
<tr>
<td>( b )</td>
<td>0.012</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>( e )</td>
<td>0.012</td>
<td>40</td>
<td></td>
</tr>
</tbody>
</table>

### TABLE 4

<table>
<thead>
<tr>
<th>(1)</th>
<th>(2)</th>
<th>(3)</th>
<th>(4)</th>
<th>(5)</th>
<th>(6)</th>
<th>(7)</th>
<th>(8)</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \Delta x )</td>
<td>( \Delta z )</td>
<td>( \Delta P )</td>
<td>( \Delta z/\Delta x )</td>
<td>( x )</td>
<td>( z )</td>
<td>( P )</td>
<td>( P_{avg} )</td>
</tr>
<tr>
<td>Initial point:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Direction: ( a )</td>
<td>-0.03</td>
<td>0.012</td>
<td>-24</td>
<td>-0.4</td>
<td>0.85000</td>
<td>0.01835</td>
<td>274</td>
</tr>
<tr>
<td>( b )</td>
<td>-0.04</td>
<td>0.012</td>
<td>-32</td>
<td>-0.3</td>
<td>0.84000</td>
<td>0.01835</td>
<td>266</td>
</tr>
<tr>
<td>( e )</td>
<td>-0.06</td>
<td>0.012</td>
<td>-40</td>
<td>-0.2</td>
<td>0.82000</td>
<td>0.01835</td>
<td>258</td>
</tr>
</tbody>
</table>

### TABLE 5A

<table>
<thead>
<tr>
<th>Node</th>
<th>( P_{avg} )</th>
<th>( x_0 )</th>
<th>( y_0 )</th>
<th>( k_1 )</th>
<th>( k_2 )</th>
<th>( k_3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( a )</td>
<td>286</td>
<td>0.88000</td>
<td>0.11365</td>
<td>16.93</td>
<td>41.0</td>
<td>1.2594</td>
</tr>
<tr>
<td>( b )</td>
<td>282</td>
<td>0.88000</td>
<td>0.11365</td>
<td>16.81</td>
<td>40.0</td>
<td>1.2594</td>
</tr>
<tr>
<td>( e )</td>
<td>278</td>
<td>0.88000</td>
<td>0.11365</td>
<td>16.70</td>
<td>39.0</td>
<td>1.2594</td>
</tr>
</tbody>
</table>

### TABLE 5B

<table>
<thead>
<tr>
<th>Node</th>
<th>( k_{x,y} )</th>
<th>( -k_{1,x} )</th>
<th>( k_{3,y} )</th>
<th>( k_{1,x} + k_{3,y} )</th>
<th>( \frac{\Delta x}{\Delta x} )</th>
<th>( k_{3,y} )</th>
<th>( \frac{\Delta x}{\Delta x} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( a )</td>
<td>4.64</td>
<td>-14.8</td>
<td>0.1433</td>
<td>-14.66</td>
<td>-0.3150</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( b )</td>
<td>4.53</td>
<td>-14.7</td>
<td>0.1433</td>
<td>-14.56</td>
<td>-0.3110</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( e )</td>
<td>4.41</td>
<td>-14.6</td>
<td>0.1433</td>
<td>-14.46</td>
<td>-0.3050</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
column 5 of Table 5B. We observe for this case that the $\Delta z/\Delta x$ values calculated from the average pressure between the nodes do not agree with the $\Delta z/\Delta x$ specified by the grid, column 4, Table 4. This means that the $x$, $z$, and $P$ at the $a$, $b$, and $e$ nodes are not physically consistent. The proper values of the pressures consistent with the $x$, $z$ composition at nodes $a$, $b$, and $e$ are 370, 254, and 122, respectively.

\[
\begin{align*}
0.82000, & 0.1835, 258 \\
0.84000, & 0.1835, 266 \\
0.85000, & 0.1835, 274 \\
0.88000, & 0.0635, 298 \\
\end{align*}
\]

**FIG. 3.** $x$, $z$, $P$ grid.

In the attempt to define uniquely each point in the three-dimensional grid, we have used up all the degrees of freedom. If, on the other hand, we specify a priori, a two-dimensional $xz$ grid and permit $v_1$ to vary continuously, we can in general find a $v_1$ that satisfies Eq. (7). This assumes, of course, that we start at a known $x_0$, $z_0$, $(v_1)_0$. We recognize, of course, that even though $v_1$ may vary continuously, there will be some compositions that are unattainable.

c. Case II

**STATE OF THE SYSTEM SPECIFIED BY $x$, $z$, $v_1$, $v_2$, ..., $v_k$**

In this system, the coefficients $k_i$ of Eq. (2) are functions of the manipulated variables $v_1$, $v_2$, ..., $v_k$.

\[
k_i = k_i(v_1, v_2, ..., v_k) \quad (8)
\]

The difficulties associated with satisfying physically the transition from one node to the adjacent nodes in a predetermined three-dimensional grid are multiplied many times over for the predetermined grid of dimension $(k + 2)$. In Case II, given a known $x_j$, $z_j$, $(v_1)_j$, $(v_2)_j$, ..., $(v_k)_j$ to drive the process to $x_{j+1}$, $z_{j+1}$, there are many combinations of values for the $(v_1)_{j+1}$, $(v_2)_{j+1}$, ..., $(v_k)_{j+1}$. The multiplicity of com-
9. EXAMPLE: CONTROL OF BATCH REACTION

Combinations of values for the $v_i$ variables coupled with setting up a predetermined grid in $(k + 2)$ space, which represent physically realizable states, complicates the numerical solution to Case II.

We must now seek other avenues of approach to reduce the dimensionality or at least to reduce the computation some way. Since we wish to steer the process from a known composition $(x_0, z_0)$, to a final composition $(x_f, z_f)$ in minimum time, it is convenient to think in terms of an evenly spaced $xz$ grid. Rather than set up a predetermined grid in $(k + 2)$ dimensions, let us fix only the $xz$ grid, and develop the $(k + 2)$-dimensional grid as we move along. We know from observations in the real world that it is not possible to permit the $v_i$ variables to assume all possible combinations of values at will, even though this is possible mathematically. A certain period of time is required to change set points of instruments, for valves to respond to changes in signals, and for the process to respond to the new conditions and settle down. It seems quite reasonable therefore, to say that each variable $v_i$ will be constrained from node to node by equations of the type

$$| (v_i)_j - (v_i)_{j+1} | \leq L_i$$

That is to say, the absolute difference between the value of the $i$th variable at node $j$ and node $(j + 1)$ is equal to or less than a constant $L_i$.

By permitting the $v_i$ variables to vary continuously within their allowed incremental range, we accomplish two things. First, we establish the $(k + 2)$-dimensional grid as we move along so that each node in $x$, $z$, $v_1$, $v_2$, ..., $v_k$ is a physically realizable state. Second, we restrict by the permissible incremental range on the $v_i$ terms, the combinations of the values of the $v_i$ terms, that we will use to drive the process from the known $x$, $z$, $v_1$, $v_2$, ..., $v_k$ node to the next $xz$ composition. Pursuing this approach we develop a $(k + 2)$-dimensional grid with unevenly spaced nodes. Along each permitted path of the grid we can compute the time. In a manner similar to Eqs. (4) and (5), we may set up the functional equations and solve the time minimization problem.

To supplement this approach, we may reduce the dimensionality by a study of the physical and chemical problem. From this study it may turn up that certain variables must always be held at their highest or lowest values. This in effect eliminates the variable from consideration in the optimization. We also may note that certain variables vary monotonically with time. This, in effect, reduces the number of possible combinations of values for these variables.

To reduce the dimensionality we may run parametric case histories and from it build toward a generalized solution. By this approach we
hold, arbitrarily, certain \( v_i \) terms at fixed levels and develop the grid and the minimum time path using the remaining \( v_i \) variables. Repetition of this technique for a range of levels for the chosen \( v_i \) variables can be used to establish the time minimization as a function of the level of the variables. Another approach which can be used to reduce the computation would be to set arbitrarily for certain \( v_i \) variables the values that they will assume as a function of time or as a function of the \( x \) composition. The remaining \( v_i \) terms would be varied within their permitted ranges and the \( (k + 2) \) grid developed step-by-step. This technique is the "approximation in policy space." The principal problems in approximating in policy space are convergence and rate of convergence. We want to generate from the results of the first approximation policy sufficient information to make the next policy approximation yield a smaller minimum time and so on, and we want to be sure that we approach the absolute minimum. Another method for reducing dimensionality, previously discussed and to be discussed further in Sections 19-22, is the use of Lagrangian multipliers.

d. Case III

**State of the System Specified by** \( x_1, x_2, \ldots, x_{m-1} \) **Composition Coordinates and One Manipulated Variable** \( v_1 \)

In some chemical systems the reactions are described by equations of the type:

\[
X_1 \Leftarrow X_2 + X_3; \quad X_2 \Leftarrow X_4 + X_5; \quad X_3 \Leftarrow X_6 + X_7 + X_8
\]

For this case we consider the reaction velocity constants \( k_i \) to be a function of only one variable, \( v_1 \) and \( \sum_{i=1}^{m} x_i = 1 \).

Since we are interested in driving the process from a known initial composition to a final composition, it is convenient to employ a predetermined \( (m - 1) \)-dimensional grid for the composition coordinates. By permitting the \( v_1 \) to vary continuously, we can generate the \( m \)-dimensional grid in \( (m - 1) \) composition variables and one manipulated variable. This guarantees that nodes are physically realizable. The time of travel between adjacent nodes and the time minimization may be carried out as in Case I.

e. Case IV

**State of the System Specified by** \( x_1, x_2, \ldots, x_{m-1} \) **Composition Coordinates** \( v_1, v_2, \ldots, v_k \) **Manipulated Variables**

The chemical system in Case IV, where the reaction velocity constants are functions of \( k \) manipulated variables, presents us with the most
10. Allocation Problems, Dynamic Programming, Differential Calculus

In view of the discussion and examples cited above, one is tempted to ask: "Why not solve the allocation problem by differential calculus?" Well, why not? Let us set up one of the examples and see what happens. Let us consider the example in Chapter 3, Section 2 for the parallel reactors.

To be specific, the profit equation is written for a 3 reactor system.

\[ P = g_1(A_1, y_1, T_1) + g_2(A_2, y_2, T_2) + g_3(A_3, y_3, T_3) \]  

(1)

The object is to maximize \( P \)

\[ P = \sum_{i=1}^{3} g_i(A_i, y_i, T_i) \]  

(2)

subject to the constraints

\[ \sum_{i=1}^{3} y_i = x \]  

(3)

\[ T_{\text{Min}} \leq T_i \leq T_{\text{Max}}, \quad i = 1, 2, 3 \]  

(4)
We may eliminate the inequality constraints in Eq. (4) by the following device:

\[ u_i^2 = (T_{\text{Max}} - T_i)(T_i - T_{\text{Min}}) \]  

Using Lagrangian multipliers, we may maximize the following expression:

\[ \Pi = \sum_{i=1}^{3} g_i(A_i, y_i, T_i) + \lambda_1 \left( \sum_{i=1}^{3} y_i - x \right) + \sum_{i=1}^{3} \eta_i \left[ u_i^2 - (T_{\text{Max}} - T_i)(T_i - T_{\text{Min}}) \right] \]  

where \( \lambda_1 \) and \( \eta_i \) are Lagrangian multipliers to be determined. Forming the partial derivatives with respect to the independent variables, we have

\[ \frac{\partial g_1}{\partial y_1} + \lambda_1 = 0 \]  
\[ \frac{\partial g_2}{\partial y_2} + \lambda_1 = 0 \]  
\[ \frac{\partial g_3}{\partial y_3} + \lambda_1 = 0 \]  
\[ \frac{\partial g_1}{\partial T_1} + \eta_1(2T_1 - T_{\text{Max}} - T_{\text{Min}}) = 0 \]  
\[ \frac{\partial g_2}{\partial T_2} + \eta_2(2T_2 - T_{\text{Max}} - T_{\text{Min}}) = 0 \]  
\[ \frac{\partial g_3}{\partial T_3} + \eta_3(2T_3 - T_{\text{Max}} - T_{\text{Min}}) = 0 \]  
\[ \eta_1 u_1 = 0 \]  
\[ \eta_2 u_2 = 0 \]  
\[ \eta_3 u_3 = 0 \]

To these equations, we must add the constraint equations

\[ y_1 + y_2 + y_3 = x \]  
\[ u_1^2 - (T_{\text{Max}} - T_1)(T_1 - T_{\text{Min}}) = 0 \]  
\[ u_2^2 - (T_{\text{Max}} - T_2)(T_2 - T_{\text{Min}}) = 0 \]  
\[ u_3^2 - (T_{\text{Max}} - T_3)(T_3 - T_{\text{Min}}) = 0 \]
We, therefore, have for the 3 reactor system, 13 equations in 13 unknowns, Eq. (7)-(19). The unknowns are $y_1, y_2, y_3, T_1, T_2, T_3, u_1, u_2, u_3, \eta_1, \eta_2, \eta_3,$ and $\lambda$. The solution of this set of simultaneous equations will yield the optimal values of $y_1, y_2, y_3,$ and $T_1, T_2, T_3$. To determine the optimum, the solution of this set of equations requires examining the equations of the form $\eta_i u_i = 0$ and solving Eqs. (7)-(19) for the three possibilities $\eta_i = 0$ and $u_i \neq 0$, $\eta_i \neq 0$ and $u_i = 0$, and $\eta_i = 0$ and $u_i = 0$. If $u_i = 0$, then $T_i$ must equal $T_{\text{Max}}$ or $T_{\text{Min}}$. If $u_i \neq 0$, then $T_i$ lies within the constraints.

Equation (6) is evaluated for each of these possibilities for $\eta_i$ and $u_i$ for $i = 1, 2, 3$. By a process of exhausting all combinations of $\eta_i$ and $u_i$, the maximum value of Eq. (6), and hence of Eq. (2), is found.

The conclusion that we may draw from this example is that Lagrangian multipliers and the differential calculus may indeed be used to solve these problems. The question of the ease of solving the equations with the various combinatorial possibilities of $\eta_i$ and $u_i$ to be examined is something else. It is at this point that dynamic programming offers relief. Rather than dealing with large sets of simultaneous equations, we can, by means of dynamic programming, solve the problem by dealing only with the successions of a single state variable (see Chapter 3, Section 2). In the differential calculus method, each constraint adds two equations, one for the constraint, the other for the associated Lagrangian multiplier. In the dynamic programming method, each constraint narrows the feasible region and in effect makes the problem easier to solve.

11. Successive Approximations, Picard's Method

One of the most powerful tools in analysis is the method of successive approximations. Before we enter into a discussion of successive approximation in dynamic programming, let us review Picard's method, the classical approach. A typical successive approximation problem is given below.

Let us consider the differential equation

$$\frac{dy}{dx} = u(x, y) \tag{1}$$

where we wish to find the curve that satisfies this equation and passes through the point $x = a, y = b$.

The solution may be obtained by successive approximations in which a sequence of functions $y_0(x), y_1(x), y_2(x), \ldots$ is generated to satisfy the boundary condition and whose limit will satisfy Eq. (1). The procedure
is as follows. An initial approximation \( y_0(x) \) is arbitrarily chosen. It is substituted into the right-hand side of Eq. (1), which is integrated to give the next approximation \( y_1(x) \). The function \( y_1(x) \) in turn is substituted into the right-hand side of Eq. (1), which is integrated to give \( y_2(x) \), and so on. Under quite simple and usable conditions, the process can be proved to converge to a function \( y(x) \) which is independent of the initial guess \( y_0(x) \).

A recursion formula for the process described is

\[
y_n(x) = b + \int_a^x u(x, y_{n-1}(x)) \, dx
\]

(2)

To illustrate the method, consider the equation

\[
\frac{dy}{dx} = 3x - \frac{y}{x}
\]

(3)

where the curve must pass through the point \( y = 2, x = 1 \), corresponding to the values \( b = 2, a = 1 \) in Eq. (2).

Let the initial approximation be

\[
y_0 = x^2
\]

(4)

Substituting Eq. (4) into Eq. (2), we have

\[
y_1(x) = 2 + \int_1^x \left( 3x - \frac{y_0(x)}{x} \right) \, dx = 2 + \int_1^x \left( 3x - \frac{x^2}{x} \right) \, dx
\]

\[
y_1(x) = 1 + x^2
\]

(5)  (6)

Substituting Eq. (6) into Eq. (2), we have

\[
y_2(x) = 2 + \int_1^x \left( 3x - \frac{y_1(x)}{x} \right) \, dx = 2 + \int_1^x \left( 3x - \frac{1 + x^2}{x} \right) \, dx
\]

\[
y_2(x) = 1 + x^2 - \ln x
\]

(7)  (8)

Continuing, we have

\[
y_3(x) = 2 + \int_1^x \left( 3x - \frac{y_2(x)}{x} \right) \, dx = 2 + \int_1^x \left( 3x - \frac{1 + x^2 - \ln x}{x} \right) \, dx
\]

\[
y_3(x) = 1 + x^2 - \ln x + \frac{1}{2} (\ln x)^2
\]

(9)  (10)

\[
y_4(x) = 1 + x^2 - \ln x + \frac{1}{2} (\ln x)^2 - \left( \frac{1}{3} \right) \left( \frac{1}{3} \right) (\ln x)^3
\]

(11)

\[
y_5(x) = 1 + x^2 - \ln x + \frac{1}{2} (\ln x)^2 - \left( \frac{1}{3} \right) \left( \frac{1}{3} \right) (\ln x)^3 + \left( \frac{1}{4} \right) \left( \frac{1}{4} \right) (\ln x)^4
\]

(12)

Proceeding this way, a pattern emerges.
Now let us recall that

$$e^v = 1 + v + \frac{v^2}{2!} + \frac{v^3}{3!} + \frac{v^4}{4!} + \ldots + \frac{v^n}{n!} + \ldots$$  \hspace{1cm} (13)$$

$$e^{-v} = 1 - v + \frac{v^2}{2!} - \frac{v^3}{3!} + \frac{v^4}{4!} - \ldots + \frac{v^n}{n!} + \ldots$$  \hspace{1cm} (14)$$

If we let

$$v = \ln x$$  \hspace{1cm} (15)$$

we observe that

$$e^{-v} = e^{-\ln x} = 1 - \ln x + \frac{(\ln x)^2}{2!} - \frac{(\ln x)^3}{3!} + \ldots + \frac{(\ln x)^n}{n!} + \ldots$$  \hspace{1cm} (16)$$

Comparing Eqs. (12) and (16), we may infer that as $$n \to \infty$$

$$y_n(x) = x^2 + e^{-\ln x} = x^2 + \frac{1}{x}$$  \hspace{1cm} (17)$$

If we substitute $$y = x^2 + 1/x$$ into the original Eq. (3), we find that it satisfies the original expression.

We have shown here that by means of an explicit calculation that the limiting form of the approximation yields the solution to the original expression. The limiting solution can be found approximately after a number of approximations no matter how good or bad the initial guess $$y_0(x)$$ was. A good initial guess, of course, yields an answer faster.

12. Methods of Successive Approximations, Dynamic Programming

In dynamic programming, problems often may be solved by methods of successive approximation. Dynamic programming provides not only the usual method of approximating, namely, approximating in function space, but it also provides the tool of approximation in policy space.

Let us use the functional equation

$$f(x) = \operatorname{Max}_{0 \leq y \leq x} [g(y) + h(x - y) + f(ay + b(x - y))]$$  \hspace{1cm} (1)$$

to distinguish these two approximations. When we approximate in function space, we guess a value of $$f(x)$$ and then determine the value of $$y$$ that maximizes the right-hand side. From this maximization, we get a better estimate of $$f(x)$$ as discussed in Section 14.
When we approximate in policy space, we guess a value of $y(x)$, substitute it in the expression, and then use the expression to get a better estimate of $y(x)$.

The approximation in policy space is many times a more natural operation, since we may have a "feel" for the right thing to do from practical experience. The actual return from the process may not be as important as knowing the proper way to guide the process.

A sensitivity analysis of approximation in policy space can be extremely meaningful. If small changes in policy cause erratic or unusual performance in function space, this behavior deserves close scrutiny.

In the sections to follow, we set up the allocation problem in several different ways. First, we approximate in policy space. We next approximate in function space, and finally we solve the problem as a discrete problem. We show that in all these formulations that the solutions tend to the same limit as the number of stages of approximation becomes large.

13. Approximation in Policy Space

A powerful tool in dynamic programming is the use of methods of successive approximation. In some problems, it is advantageous to approximate in policy space. By that we mean that we assume a policy for the decision variables and then compute the return. We next take the computed return and use it to generate the next policy approximation. From the next policy approximation, the return function is computed. The process is repeated over and over again, until after $(n + 1)$ repetitions the $(n + 1)$th return function based on the $(n + 1)$th policy approximation does not differ materially from the $n$th return function based on the $n$th policy approximation.

To illustrate the idea, let us consider the following equation for an infinite stage return function $f(x)$:

$$f(x) = \max_{0 \leq y \leq x} [g(y) + h(x - y) + f(ay + b(x - y))], \quad 0 < a, \quad b < 1$$

This return function arises in processes where a resource $x$ must be divided into two parts, one of which gives a yield $g(y)$ and the other gives a yield of $h(x - y)$. In the course of the process, the quantity $y$ is reduced to $ay$ and the quantity $(x - y)$ is reduced to $b(x - y)$. The quantity $y$ is chosen to maximize the right-hand side of the equation.

In the procedure about to be developed, the notation $y_n(x)$ and $f_n(x)$ will be used to denote the $n$th approximation of $y$ and $f(x)$. Here we use a multi-stage approximation to generate the infinite stage policy.
13. APPROXIMATION IN POLICY SPACE

Let us assume for simplicity, that the return functions are

$$ g(y) = \sqrt{y} \quad (2) $$
$$ h(x - y) = x - y \quad (3) $$

Let the initial approximation of $y$ for all $x$ be

$$ y_0(x) = 0 \quad (4) $$

Then Eq. (1) takes the form

$$ f_0(x) = x + f_0(bx) \quad (5) $$

Using Eq. (5), $f_0(bx)$ can be written as

$$ f_0(bx) = bx + f_0(b^2x) \quad (6) $$

Repeated application of Eq. (5) yields for $f_0(x)$ the following equation:

$$ f_0(x) = x + bx + b^2x + ... = \frac{x}{1 - b} \quad (7) $$

In order to improve the initial trial value of $y$, we form $f_1(x)$ and maximize over $y$ to obtain

$$ f_1(x) = \max_{0 \leq y \leq x} \left[ \sqrt{y} + (x - y) + f_0(ay + b(x - y)) \right] \quad (8) $$

Substituting $f_0(x)$ from Eq. (7) with the argument $x = ay + b(x - y)$ into Eq. (8), we have

$$ f_1(x) = \max_{0 \leq y \leq x} \left[ \sqrt{y} + (x - y) + \frac{ay + b(x - y)}{1 - b} \right] \quad (9) $$

The maximum in Eq. (9) can be found by forming the derivative with respect to $y$ and setting it equal to zero, provided that the maximum value lies in the range $(0, x)$. Otherwise, the maximum may be at $y = 0$ or $x$.

$$ \frac{1}{2} \frac{1}{\sqrt{y}} - 1 + \frac{a - b}{1 - b} = 0 \quad (10) $$

The maximizing $y$ is

$$ y_1(x) = \frac{1}{4} \left( \frac{1 - b}{1 - a} \right)^2 \quad \text{provided that we assume} \quad x > \frac{1}{4} \left( \frac{1 - b}{1 - a} \right)^2 \quad (11) $$
The corresponding return function is given by

$$f_1(x) = \frac{1}{4} \left( \frac{1 - b}{1 - a} \right) + \frac{x}{1 - b} \quad (12)$$

Continuing,

$$f_2(x) = \max_{0 \leq y \leq x} \left[ \sqrt{y} + (x - y) + f_1(ay + b(x - y)) \right] \quad (13)$$

Substituting $f_1(x)$ from Eq. (12) with the argument $x = ay + b(x - y)$ into Eq. (13), we have

$$f_2(x) = \max_{0 \leq y \leq x} \left[ \sqrt{y} + (x - y) + \frac{1}{4} \left( \frac{1 - b}{1 - a} \right) + \frac{ay + b(x - y)}{1 - b} \right] \quad (14)$$

Assuming that the maximum value of $y$ exists in the interval $(0, x)$, we take the derivative with respect to $y$ in Eq. (14) and set it equal to zero:

$$\frac{1}{2 \sqrt{y}} - 1 + \frac{a - b}{1 - b} = 0 \quad (15)$$

The maximizing value of $y$ is

$$y_2(x) = \frac{1}{4} \left( \frac{1 - b}{1 - a} \right)^2 \quad \text{provided} \quad x > \frac{1}{4} \left( \frac{1 - b}{1 - a} \right)^2 \quad (16)$$

The corresponding return function is given by

$$f_3(x) = \frac{1}{2} \left( \frac{1 - b}{1 - a} \right) + \frac{x}{1 - b} \quad (17)$$

The third approximation is found from

$$f_4(x) = \max_{0 \leq y \leq x} \left[ \sqrt{y} + (x - y) + f_2(ay + b(x - y)) \right] \quad (18)$$

Substituting Eq. (17) with the argument $x = ay + b(x - y)$ into Eq. (18), we have

$$f_3(x) = \max_{0 \leq y \leq x} \left[ \sqrt{y} + (x - y) + \frac{1 - b}{1 - a} + \frac{ay + b(x - y)}{1 - b} \right] \quad (19)$$

Taking the derivative of Eq. (19) with respect to $y$ and assuming the maximum lies within $(0, x)$, we find

$$y_3(x) = \frac{1}{4} \left( \frac{1 - b}{1 - a} \right)^2 \quad \text{provided} \quad x > \frac{1}{4} \left( \frac{1 - b}{1 - a} \right)^2 \quad (20)$$
The corresponding return function is

\[ f_3(x) = \frac{3}{4} \left(\frac{1 - b}{1 - a}\right) + \frac{x}{1 - b} \]  

(21)

Proceeding as above, we may show that for the \( k \)-stage process

\[ y_k(x) = \frac{1}{4} \left(\frac{1 - b}{1 - a}\right)^k \]  

(22)

\[ f_k(x) = \frac{k}{4} \left(\frac{1 - b}{1 - a}\right) + \frac{x}{1 - b} \]  

(23)

In Table 6, the results of the approximation are summarized. We observe that the initial guess in policy space \( y_0(x) = 0 \) leads to the next

<table>
<thead>
<tr>
<th>TABLE 6</th>
<th>SUMMARY OF APPROXIMATION IN POLICY SPACE</th>
</tr>
</thead>
<tbody>
<tr>
<td>( y_0(x) = 0 )</td>
<td>( f_0(x) = \frac{x}{1 - b} )</td>
</tr>
<tr>
<td>( y_1(x) = \frac{1}{4} \left(\frac{1 - b}{1 - a}\right)^2 )</td>
<td>( f_1(x) = \frac{1}{4} \left(\frac{1 - b}{1 - a}\right) + \frac{x}{1 - b} )</td>
</tr>
<tr>
<td>( y_2(x) = \frac{1}{4} \left(\frac{1 - b}{1 - a}\right)^2 )</td>
<td>( f_2(x) = \frac{1}{2} \left(\frac{1 - b}{1 - a}\right) + \frac{x}{1 - b} )</td>
</tr>
<tr>
<td>( y_3(x) = \frac{1}{4} \left(\frac{1 - b}{1 - a}\right)^2 )</td>
<td>( f_3(x) = \frac{3}{4} \left(\frac{1 - b}{1 - a}\right) + \frac{x}{1 - b} )</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>( y_4(x) = \frac{1}{4} \left(\frac{1 - b}{1 - a}\right)^2 )</td>
<td>( f_4(x) = \frac{k}{4} \left(\frac{1 - b}{1 - a}\right) + \frac{x}{1 - b} )</td>
</tr>
</tbody>
</table>

approximation in policy space \( y_1(x) \), which yields the same values as \( y_k(x) = \frac{1}{4}(1 - b/1 - a)^k \), \( k = 1, 2, 3, \ldots \). The return functions can be described recursively as

\[ f_k(x) = \frac{1}{4} \left(\frac{1 - b}{1 - a}\right) + f_{k-1}(x) \]  

(24)

or

\[ f_k(x) = \frac{k}{4} \left(\frac{1 - b}{1 - a}\right) + \frac{x}{1 - b} \]  

(25)
14. Approximation in Function Space

It is sometimes useful to approximate in function space. Starting with an initial function $f_0(x)$, which is assumed, the solution of the functional equation yields the next approximation $f_1(x)$. Substituting $f_1(x)$ into the functional equation and optimizing yields the next approximation $f_2(x)$. Continuing the procedure, the approximation in function space converges in the limit to the actual solution.

To illustrate the idea more explicitly, let us consider the function given in Section 13 and work out the approximation in function space. We desire to maximize

$$f(x) = \max_{0 \leq y \leq x} [g(y) + h(x - y) + f(ay + b(x - y))], \quad 0 < a, \ b < 1 \quad (1)$$

where we take

$$g(y) = \sqrt{y} \quad (2)$$
$$h(x - y) = x - y \quad (3)$$

As the initial guess in function space, let

$$f_0(x) = ax \quad (4)$$
$$f_1(x) = \max_{0 \leq y \leq x} [\sqrt{y} + (x - y) + f_0(ay + b(x - y))] \quad (5)$$
$$f_1(x) = \max_{0 \leq y \leq x} [\sqrt{y} + (x - y) + a(ay + b(x - y))] \quad (6)$$

where Eq. (4) is substituted into Eq. (5) with the argument $x$ taken as $ay + b(x - y)$.

The maximum value in Eq. (6) can be found by forming the derivative with respect to $y$ and setting it equal to zero, provided $y$ lies inside $(0, x)$. Otherwise, the maximum occurs at $y = 0$ or $x$.

$$y_1(x) = \frac{1}{4[1 - a(a - b)]^2} \quad (7)$$

Substituting Eq. (7) into Eq. (6), we find the next approximation in function space:

$$f_1(x) = \sqrt{y_1} + (x - y_1) + a(ay_1 + b(x - y_1)) \quad (8)$$
$$f_1(x) = x(1 + ab) + \frac{1}{4[1 - a(a - b)]} \quad (9)$$
Now we develop $f_2(x)$ as follows:

$$f_2(x) = \max_{0 \leq y \leq x} [\sqrt{y} + (x - y) + f_1(ay + b(x - y))]$$  \hspace{1cm} (10)$$

Substituting Eq. (9) into Eq. (10) where the argument $x$ is taken to be $ay + b(x - y)$, and differentiating Eq. (10) with respect to $y$ and setting the derivative equal to zero, we have the maximum value of $y$ as

$$y_2(x) = \frac{1}{4[1 - (1 + ab)(a - b)]^2}$$  \hspace{1cm} (11)$$

This is true provided $0 < y_2(x) < x$.

Substituting $y_2(x)$ into Eq. (10) yields $f_2(x)$:

$$f_2(x) = x[1 + b(1 + ab)] + \frac{1}{4[1 - (1 + ab)(a - b)]} + \frac{1}{4[1 - a(a - b)]}$$  \hspace{1cm} (12)$$

Continuing in this manner, we see that

$$f_k(x) = \max_{0 \leq y \leq x} [\sqrt{y} + (x - y) + f_{k-1}(ay + b(x - y))]$$  \hspace{1cm} (13)$$

In Table 7 are listed the first few terms for the successive approximations in function space and the corresponding policies. In addition, the $k$th or general approximations are given for the function and the policy.

It is interesting to compare the policy and return functions generated by the two techniques, approximation in policy space and approximation in function space. While at first glance the policy and return functions in Tables 6 and 7 appear to be dissimilar, closer scrutiny reveals that under suitable conditions, the results of the two approximation techniques approach one another.

To see this, let us first observe that the term $\sum_{n=0}^{k-1} b^n$ may be written

$$S_k = \frac{b^k - 1}{b - 1}$$  \hspace{1cm} (14)$$

As $k \to \infty$, if $|b| < 1$ the summation expression reduces to

$$S_k = \frac{1}{1 - b}$$  \hspace{1cm} (15)$$

Let us next observe that such terms as $ab^k$ or $ab^{k-1}$ tend toward zero as $k$ becomes large, since $0 < a, b < 1$.

If we apply these two observations to $y_k(x)$ in Table 7, we see that for large $k$

$$y_k(x) \approx \frac{1}{4} \left(\frac{1 - b}{1 - a}\right)^2$$  \hspace{1cm} (16)$$
### TABLE 7

**SUMMARY OF APPROXIMATION IN FUNCTION SPACE**

<table>
<thead>
<tr>
<th>$y_1(x)$</th>
<th>$f_1(x) = ax$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$y_2(x) = \frac{1}{4[1 - a(a - b)]^2}$</td>
<td>$f_2(x) = (1 + ab) x + \frac{1}{4[1 - a(a - b)]}$</td>
</tr>
<tr>
<td>$y_3(x) = \frac{1}{4[1 - (1 + ab) (a - b)]^2}$</td>
<td>$f_3(x) = [1 + b(1 + ab)] x + \frac{1}{4[1 - (1 + ab) (a - b)]} + \frac{1}{4[1 - a(a - b)]}$</td>
</tr>
<tr>
<td>$y_4(x) = \frac{1}{4[1 - (1 + b)(1 + ab)] (a - b)]^2}$</td>
<td>$f_4(x) = [1 + b(1 + b(1 + ab))] x + \frac{1}{4[1 - (1 + b)(1 + ab)] (a - b)] + \frac{1}{4[1 - a(a - b)]}$</td>
</tr>
<tr>
<td>$y_5(x) = \frac{1}{4[1 - {\sum_{n=0}^{k-2} b^n + ab^{k-1}} (a - b)]^2}$</td>
<td>$f_5(x) = \left[\sum_{n=0}^{k-1} b^n + ab^k\right] x + \sum_{r=0}^{k-1} \frac{1}{4[1 - {\sum_{n=0}^{r-1} b^n + ab^r} (a - b)]}$</td>
</tr>
</tbody>
</table>
The approximation in function space, therefore, yields in the limit the same policy function as the approximation in policy space.

In a similar manner, we may apply these two observations to the $f_k(x)$ function in Table 7. For large $k$ many of the terms in the denominator of the second summation may be approximated by Eq. (15). As a consequence, the $f_k(x)$ expression may be written as

$$f_k(x) \approx \frac{x}{1-b} + \frac{k}{4} \frac{1-b}{1-a} \quad (17)$$

This equation approaches the return function found from the policy space approximation. We may conclude, therefore, that as the number of approximations increases, the policy and return functions from the two approximation techniques approach one another in the limit.

It is interesting to note that in the function space approximation results the term $b$ has a greater influence on the policy and return function values than $a$. This is due to the fact that $b$ appears in many places, as $b$ raised to a power. If $b$ is a small number, the number of approximations required is also small, and conversely if $b$ is close to 1, many stages will be required.

15. Confirmation of Previous Examples

To prove that the approximation in policy and function space really work, let us look upon the same return function as an $N$-stage process. As $N$ gets large, the $N$-stage process approaches the infinite-stage process in the limit. In this development, the $N$ stages are not stages of approximation in policy space or function space as in the previous sections, but an $N$-stage process as we have discussed it throughout most of this book.

The return function over $N$ stages is

$$f_N(x) = \max_{0 \leq y \leq x} \left[ g(y) + h(x - y) + f_{N-1}(ay + b(x - y)) \right] \quad (1)$$

$$f_1(x) = \max_{0 \leq y \leq x} \left[ g(y) + h(x - y) \right] \quad (2)$$

Letting

$$g(y) = \sqrt{y} \quad (3)$$

$$h(x - y) = x - y \quad (4)$$
we substitute Eqs. (3) and (4) into Eqs. (1) and (2):

\[ f_N(x) = \text{Max}_{0 \leq y \leq x} \left[ \sqrt{y} + (x - y) + f_{N-1}(ay + b(x - y)) \right] \]  
\[ f_1(x) = \text{Max}_{0 \leq y \leq x} \left[ \sqrt{y} + (x - y) \right] \]  

Taking the derivative with respect to \(y\) of the right-hand side of Eq. (6) and setting it equal to zero we find

\[ y_1 = \begin{cases} \frac{1}{4}, & x > \frac{1}{4} \\ x, & 0 \leq x \leq \frac{1}{4} \end{cases} \]  

By the use of Eq. (7), the return function \(f_1(x)\) equals

\[ f_1(x) = \begin{cases} x + \frac{1}{4}, & x > \frac{1}{4} \\ \sqrt{x}, & 0 \leq x \leq \frac{1}{4} \end{cases} \]  

For the two-stage process, the \(f_2(x)\) is expressed as

\[ f_2(x) = \text{Max}_{0 \leq y \leq x} \left[ \sqrt{y} + (x - y) + f_1(ay + b(x - y)) \right] \]  

But by Eq. (8) for an interior maximum

\[ f_1(ay + b(x - y)) = ay + b(x - y) + \frac{1}{4} \]  

Substituting Eq. (10) into Eq. (9), we have

\[ f_2(x) = \text{Max}_{0 \leq y \leq x} \left[ \sqrt{y} + (x - y) + ay + b(x - y) + \frac{1}{4} \right] \]  

The derivative with respect to \(y\) of the right-hand side of Eq. (11) when set equal to zero yields

\[ y_2 = \begin{cases} \frac{1}{4[1 - (a - b)]^2}, & x > \frac{1}{4[1 - (a - b)]^2} \\ x, & 0 \leq x \leq \frac{1}{4[1 - (a - b)]^2} \end{cases} \]  

Substituting Eq. (12) into Eq. (11), we have

\[ f_2(x) = \begin{cases} (1 + b)x + \frac{1}{4[1 - (a - b)]} + \frac{1}{4}, & x > \frac{1}{4[1 - (a - b)]^2} \\ \sqrt{x} + ax + \frac{1}{4}, & 0 \leq x \leq \frac{1}{4[1 - (a - b)]} \end{cases} \]
For a three stage process, \( f_3(x) \) is expressed as
\[
f_3(x) = \max_{0 \leq y \leq x} \left[ \sqrt{y} + (x - y) + f_4(ay + b(x - y)) \right]
\] (14)

Using the results of Eq. (13) for an interior maximum, we write \( f_3(x) \) as
\[
f_3(x) = \max_{0 \leq y \leq x} \left[ \sqrt{y} + (x - y) + (1 + b) \{ ay + b(x - y) \} + \frac{1}{4[1 - (a - b)]} + \frac{1}{4[1 - (1 + b) (a - b)]} \right]
\] (15)

The optimal value of \( y \) is
\[
y_3 = \begin{cases} 
\frac{1}{4[1 - (1 + b) (a - b)]^2}, & x > \frac{1}{4[1 - (1 + b) (a - b)]^2} \\
x, & 0 \leq x \leq \frac{1}{4[1 - (1 + b) (a - b)]^2}
\end{cases}
\] (16)

The return for the three-stage process is given by
\[
f_3(x) = \begin{cases} 
[1 + (1 + b) b] x + \frac{1}{4[1 - (1 + b) (a - b)]} + \frac{1}{4[1 - (a - b)]} + \frac{1}{4}, & x > \frac{1}{4[1 - (1 + b) (a - b)]^2} \\
\sqrt{x} + (1 + b) ax + \frac{1}{4[1 - (a - b)]} + \frac{1}{4}, & 0 \leq x \leq \frac{1}{4[1 - (1 + b) (a - b)]^2}
\end{cases}
\] (17)

Continuing in this manner, we develop Table 8.

Comparing the results of the \( k \)-stage process in Table 8 with the results of the approximations in policy space and function in Tables 6 and 7, we observe that, as \( k \) increases, the policies and return functions for all three methods approach one another in the limit.

16. Quasi-Optimal Policy Approximation

If we consider again the infinite stage allocation problem
\[
f(x) = \max_{0 \leq y \leq x} [g(y) + h(x - y) + f(ay + b(x - y))]
\] (1)

we may develop an approximation policy as follows. We write
\[
f(ay + b(x - y)) = f(bx + (a - b) y)
\] (2)
TABLE 8

N-Stage Process Approximation

<table>
<thead>
<tr>
<th>n</th>
<th>N = 1</th>
<th>N = 2</th>
<th>N = 3</th>
<th>N = 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>x + 1/4</td>
<td>1/(1 + b) x + 1/(4(1 - (a - b))] + 1</td>
<td>1/(4(1 - (a - b))] + 1</td>
<td>1/(4(1 - (a - b))] + 1</td>
</tr>
<tr>
<td>2</td>
<td>4(1 - (a - b)]^2</td>
<td>4(1 - (a - b)]^2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>b/2</td>
<td>b/2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>(\sum_{k=1}^{(a - b]^2} )</td>
<td>(\sum_{k=1}^{(a - b]^2} )</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[ f_k(x) = \frac{1}{4} \left(1 + \frac{b}{(a - b)^2}\right) \]
Adding and subtracting $x$ to the right hand side of (2), we write

$$f(ay + b(x-y)) = f[(1 - (1 - b)) x + (a - b) y] = f[x - (1 - b) x + (a - b) y]$$

(3)

If we expand by a Taylor's series around $x$ over the increment $[-(1 - b) x + (a - b) y]$, we have

$$f(x - (1 - b) x + (a - b) y) = f(x) + [- (1 - b) x + (a - b) y] f'(x) + ... + ...$$

(4)

On substituting Eq. (4) into Eq. (1), we find

$$f(x) = \max_{0 \leq y \leq x} \left[ g(y) + h(x - y) + f(x) + \{- (1 - b) x + (a - b) y\} f'(x) \right]$$

(5)

$$0 = \max_{0 \leq y \leq x} \left[ g(y) + h(x - y) + \{- (1 - b) x + (a - b) y\} f'(x) \right]$$

(6)

$$f'(x) = \max_{0 \leq y \leq x} \left[ \frac{g(y) + h(x - y)}{(1 - b) x - (a - b) y} \right]$$

(7)

Rearranging, we express Eq. (7) as

$$f'(x) = \max_{0 \leq y \leq x} \left[ \frac{g(y) + h(x - y)}{x - \{ ay + b(x - y)\}} \right]$$

(8)

From Eq. (8) the policy determined here may be stated as maximizing the ratio of the immediate return, $g(y) + h(x - y)$ to the immediate consumption or expenditure of the resource, $x - \{ ay + b(x - y)\}$.

If we use as before $g(y) = \sqrt{y}$ and $h(x - y) = (x - y)$, we write Eq. (8) as

$$f'(x) = \max_{0 \leq y \leq x} \left[ \frac{\sqrt{y} + (x - y)}{x - \{ ay + b(x - y)\}} \right]$$

(9)

On differentiating the right-hand side of Eq. (9) with respect to $y$, and setting the derivative equal to zero, we find that policy for $y$ is determined by

$$y + \frac{2(a - 1) x \sqrt{y}}{a - b} + \frac{(1 - b)}{(a - b)} x = 0$$

(10)

Solving Eq. (10), which is a quadratic in $\sqrt{y}$, for $y$ yields

$$y = 2 \left( \frac{a - 1}{a - b} \right)^2 x^2 \pm \frac{2(a - 1)}{(a - b)^2} \sqrt{(a - 1)^2 x^2 - (1 - b)(a - b) x}$$

(11)
If we compare this quasi-optimal policy with those developed in Sections 13–15, we discover that the quasi-optimal policy is the only one dependent on \( x \).

17. Alternative Policy Space Approximation

In Section 13, we outlined the idea of approximation in policy space and gave an example using an initial policy \( y_0(x) = 0 \). We may approximate in policy space by a variety of different policies. Referring to the allocation problem of Section 13, we list some possibilities:

(a) We may arbitrarily choose an initial policy, \( y_0(x) = s(x) \), where \( s(x) \) is a known function of \( x \).

(b) We may choose a policy to maximize the immediate return \( g(y) + h(x - y) \).

(c) We may choose \( y \) so that

\[
\frac{g(y)}{(1 - a) y} = \frac{h(x - y)}{(1 - b)(x - y)}
\]

(d) We may choose \( y \) to maximize

\[
\frac{g(y) + h(x - y)}{x - (ay + b(x - y))}
\]

This is the result of the quasi-optimal policy given in Section 16.

(e) We may approximate \( g(y) \) and \( h(x - y) \) by quadratic polynomials and use the exact solution of the quadratic formulation to generate a policy.

In item (e), the quadratic approximations of the functions \( g(y) \) and \( h(x - y) \) may be formed by any one of a number of curve fitting techniques. Polynomial approximations and least squares fitting, employing orthogonal or nonorthogonal functions, are typical of these techniques.

In view of the importance of item (e), we summarize below the exact solution of the quadratic form of the allocation problem.

If \( c, d > 0 \) and \( 0 < b \leq a < 1 \), and if \( g(y) = cy - y^2 \) and \( h(x - y) = d(x - y) - (x - y)^2 \), then the \( f(x) \) function may be expressed as

\[
f(x) = \Max_{0 \leq y \leq x} [cy - y^2 + d(x - y) - (x - y)^2 + f(ay + b(x - y))]
\]

(1)

with the initial condition \( f_0 = 0 \).
In the interval $0 \leq x \leq \text{Min}(c/2, d/2)$, $f(x)$ takes on three forms depending on the sign of $(c/1 - a) - (d/1 - b)$.

**Case I:**

$$f(x) = \frac{c}{1-a} x - \frac{d}{1-b}$$

where

$$\alpha = \left(1 + \frac{1}{2} \left(\frac{a^2 - b^2}{a - b}\right)\right) + \sqrt{1 + \frac{1}{4} \left(\frac{a^2 - b^2}{1-ab}\right)^2}^{-1}$$

**Case II:**

$$f(x) = \left(\frac{d}{1-b}\right) x - \left(\frac{1}{1-b^2}\right) x^2$$

for $0 \leq x \leq \text{Min} \left[\lambda, \frac{c}{2}, \frac{d}{2}\right]$ where

$$\lambda = \frac{(1 + b) [d(1-a) - c(1-b)]}{2(1-ab)}$$

**Case III:**

$$f(x) = \left(\frac{c}{1-a}\right) x - \left(\frac{1}{1-a^2}\right) x^2$$

for $0 \leq x \leq \text{Min} \left[\mu, \frac{c}{2}, \frac{d}{2}\right]$ where

$$\mu = \frac{(1 + a) [c(a-b) - d(1-a)]}{2(1-ab)}$$

As we may observe, there are numerous ways to approximate in policy space. Certainly, any information about the process that the engineer possesses can be used to good stead in selecting an initial policy.

18. Analytical Return Functions

It has been shown for certain simple analytical functions with certain properties such as concavity or convexity, the structure of the solution of

$$f_N(x) = \max_{0 \leq y \leq x} \left[g(y) + h(x - y) + f_{N-1}(ay + b(x - y))\right]$$

can be deduced [16].
First, let us point out that a convex function $\varphi(x)$ looks like the curve in Fig. 4. It has the property that a maximum occurs only at either extremity. Algebraically, a convex function may be determined by testing the condition

$$\varphi''(x) > 0 \quad (2)$$

For example

$$\varphi(x) = 5 - 3x + x^2, \quad x > 0 \quad (3)$$

satisfies this criterion.

A concave function $\Phi(x)$ looks like the curve in Fig. 5. It has its
maximum somewhere between the end points. Algebraically, a concave function may be determined by testing the condition

\[ \Phi''(x) < 0 \]  \hspace{1cm} (4)

For example,

\[ \Phi(x) = -5 + 3x - x^2 \]  \hspace{1cm} (5)

With reference to Eq. (1), Bellman’s results are summarized below:

(1) If \( g(x) \) and \( h(x) \) are both convex functions of \( x \), an optimal policy requires that \( y = 0 \), or \( y = x \). In addition, \( f(x) \) is also convex. This result holds for the continuous version.

(2) If \( g(x) \) and \( h(x) \) are both strictly concave and if the following conditions are satisfied:

\[
\begin{align*}
&g'(x) > 0, \quad x > 0 \\
&h'(x) < 0, \quad x > 0 \\
&g''(x) < 0, \quad x > 0 \\
&h''(x) < 0, \quad x > 0
\end{align*}
\]

then at each stage \( k \) of the \( N \)-stage process there is a unique \( y_k = y_k(x) \) that yields the maximum. In addition, \( f(x) \) is strictly concave.

If \( b \leq a \), then \( y_1 \leq y_2 \leq y_3 \leq \ldots \leq y_N \)

If \( b \geq a \), then \( y_1 \geq y_2 \geq y_3 \geq \ldots \geq y_N \)

If \( y_k(x) = x \) and \( b \leq a \), then \( y_{k+1}(x) = x, y_{k+2}(x) = x, \ldots, y_N(x) = x \).

The advantage of these results is that the yield structure has been determined. This permits one to ascertain quickly whether the function \( g(x) \) and \( h(x) \) satisfy the criteria. From this knowledge, one knows what to expect from the numerical evaluation of the function. For example, for the concave functions and for \( b \leq a \) with \( y_k(x) = x \), there is no need to compute \( y_{k+1}(x) \) onto \( y_N(x) \) since all of the values of \( y_{k+1}(x) \) through \( y_N(x) \) equal \( x \).

In addition, for the purposes of approximating in function or policy space, the knowledge of these exact results may be used to establish useful initial guesses (see Section 17).

19. Lagrangian Multiplier

The Lagrangian multiplier is used in the calculus and the calculus of variations to solve constrained problems. The Lagrangian multiplier
can be used to solve constrained problems also in dynamic programming. A typical situation where this occurs in dynamic programming is in a problem where a fixed amount of a resource must be consumed over $N$ stages.

Let us refer to Chapter 3, Section 5 for the case of a Single Reactor, Catalyst Activity Variable. Here the cumulative feed rate over the $N$ stages of time is specified by $x$. The cumulative flow through the catalyst bed is specified by $S$. The results of Chapter 3, Section 5 are:

$$f_N(x, S_N) = \max_{0 \leq y_N \leq x} [g_N(y_N, S_N) + f_{N-1}(x - y_N, S_N + y_N)]$$  \hspace{1cm} (1)

$$f_i(x, S_i) = \max_{0 \leq y_i \leq x} [g_i(y_i, S_i)] = g(x, S_i)$$ \hspace{1cm} (2)

$$\sum_{i=1}^{N} y_i = x$$ \hspace{1cm} (3)

If the problem is not constrained by Eq. (3), the formulation is

$$f_N(S_N) = \max_{y_N} [g_N(y_N, S_N) + f_{N-1}(S_N + y_N)]$$  \hspace{1cm} (4)

$$f_i(S_i) = \max_{y_i} [g_i(y_i, S_i)]$$ \hspace{1cm} (5)

Returning to the problem where the total feed available is fixed, to reduce the state of the system from two variables to one, which is always an advantage computationally, we introduce a Lagrangian multiplier $\lambda$. The problem described by Eqs. (1)–(3) is now formulated as

$$f_N(S_N) = \max_{0 \leq y_N \leq x} [g_N(y_N, S_N) - \lambda y_N + f_{N-1}(S_N + y_N)]$$ \hspace{1cm} (6)

$$f_i(S_i) = \max_{0 \leq y_i \leq x} [g_i(y_i, S_i) - \lambda y_i] = g(x, S_i) - \lambda x$$ \hspace{1cm} (7)

The $\lambda$ is a constant over the $N$ stages and is chosen to satisfy (3).

In this formulation, the Lagrangian multiplier may be considered to be the cost of using $y_i$. For a given $\lambda$, the larger the $y_i$, the larger the total cost. A small $\lambda$ encourages the use of large $y_i$ and conversely a large $\lambda$ discourages the use of $y_i$.

Now the problem arises as to how the value of $\lambda$ is found. In the calculus and the calculus of variations, the value of the Lagrangian multiplier is determined as one of the results of the problem; that is to say, the Lagrangian multiplier is a variable just like any of the other variables in the problem.
20. LAGRANGIAN MULTIPLIER EXAMPLE

In contrast to the calculus and the calculus of variations, the Lagrangian multiplier, as used in dynamic programming, is arbitrarily chosen and held constant over the $N$ stages. If for an arbitrarily chosen $\lambda$ at the end of $N$ stages of calculation, the sum $\sum_{i=1}^{N} y_i \neq x$, then the computation is repeated for another value of $\lambda$ until it does equal $x$.

What we are doing in fact is running calculations for a range of $\lambda$ values. As a consequence of this repeated computation, we generate the optimum return on the process for a range of $\lambda$'s and their corresponding $x$'s. We are solving not only the problem at hand, but also a number of similar problems. In effect, the determination of the proper $\lambda$ is a sensitivity analysis of the problem.

It is interesting to note that this problem, set up as a calculus of variations problem, would be stated as: maximize the quantity

$$P = \int_0^T [g(y, S) - \lambda y] \, dt$$

subject to

$$\int_0^T y \, dt = x = \text{constant}$$

The quantity $\lambda$ is a constant over $(0, T)$.

The constraint in the dynamic programming problem $\sum_{i=1}^{N} y_i = x$ corresponds to the isoperimetric constraint $x = \int_0^T y \, dt$. The Lagrangian multiplier in dynamic programming corresponds to the Lagrangian multiplier in the calculus of variations, and like it, is a constant over the $N$ stages for the isoperimetric case.

20. Lagrangian Multiplier Example

To illustrate the use of the Lagrangian multiplier, let us consider the following problem based on the discussion in Chapter 3, Sections 5-6.

We desire to minimize the cost of operating a chemical reactor over three stages of time subject to the conditions listed below.

The cost function for time $k$ is:

$$g_k(y_k, S) = y_k^2 - S + 100, \quad k = 1, 2, 3$$

where

$$y_k = \text{the flow rate entering the reactor at time } k$$

$$S = \text{the cumulative flow that has passed through the reactor at the beginning of each time stage}$$
The cumulative flow to be consumed during the three time stages is 18:

\[ \sum_{i=1}^{3} y_i = y_1 + y_2 + y_3 = 18 \quad (4) \]

In time stage 1, the flow rates may assume any of the following rates:

\[ y_1 = 2, 4, 6, 8, 10 \quad (5) \]

In time stage 2, the flow rate may assume any of the following rates:

\[ y_2 = 2, 4, 6, 8, 10 \quad (6) \]

In time stage 3, the flow rate may assume any of the following rates:

\[ y_3 = 2, 4, 6 \quad (7) \]

The general functional equations are

\[ f_N(S) = \min_{\{y_N\}} [g_N(y_N, S) - \lambda y_N + f_{N-1}(y_{N-1} + S)] \quad (8) \]

\[ f_1(S) = \min_{\{y_1\}} [g_1(y_1, S) - \lambda y_1] \quad (9) \]

where \( \{y_N\} \) refers to the set of permitted rates. These are subject to

\[ \sum_{i=1}^{3} y_i = x \quad (10) \]

To be specific for the example given above

\[ f_N(S) = \min_{\{y_N\}} [y_N^2 - S + 100 - \lambda y_N + f_{N-1}(y_{N-1} + S)], \quad N = 2, 3, ... \quad (11) \]

\[ f_1(S) = \min_{\{y_1\}} [y_1^2 - S + 100 - \lambda y_1] \quad (12) \]

\[ \sum_{i=1}^{3} y_i = 18 \quad (13) \]

Equations (11)–(13) are subject to the conditions in Eqs. (5)–(7).

The results of the numerical evaluation of Eqs. (5)–(7) and (11)–(13) are listed in Tables 9A and 9B for \( \lambda = 0 \) and \( \lambda = 4 \), respectively. In each cell in Tables 9A and 9B are two entries, the \( f_k(S) \) and the corresponding \( y_k \). For example, in Table 9B starting in state \( S = 0 \) for the third stage,
$f_3(0) = 282$. The flow rate in stage 3 is equal to either 2 or 4. If the flow rate $y_3 = 4$ is chosen, then the state for stage 2 becomes $S = 0 + 4 = 4$ and $f_2(4) = 182$. For a two-stage process beginning in state $S = 4$, the optimal flow rate is $y_2 = 2$. As a consequence, the state of the system is altered to $S = 4 + 2 = 6$ for the one-stage process. In the one-stage process, $y_1 = 2$ is the optimal flow rate. The cumulative flow rate is $4 + 2 + 2 = 8$. The last line of the table gives the cumulative flow rate.

For various arbitrarily chosen $\lambda$, Table 10 gives the flow rate distribu-

**TABLE 9A**

\[
f_N(S) = \min \{ y_N^2 - S + 100 - \lambda y_N + f_{N-1}(y_N + S) \}
\]

\[\lambda = 0\]

<table>
<thead>
<tr>
<th>State</th>
<th>0</th>
<th>2</th>
<th>4</th>
<th>6</th>
<th>8</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$f_1(S)$</td>
<td>104</td>
<td>102</td>
<td>100</td>
<td>98</td>
<td>96</td>
<td>94</td>
</tr>
<tr>
<td>$y_1$</td>
<td>2</td>
<td></td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>$f_2(S)$</td>
<td>206</td>
<td>202</td>
<td>198</td>
<td>194</td>
<td>190</td>
<td></td>
</tr>
<tr>
<td>$y_2$</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$f_3(S)$</td>
<td>306</td>
<td>300</td>
<td>294</td>
<td>288</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$y_3$</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\sum_{i=1}^3 y_i$</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**TABLE 9B**

\[\lambda = 4\]

<table>
<thead>
<tr>
<th>State</th>
<th>0</th>
<th>2</th>
<th>4</th>
<th>6</th>
<th>8</th>
<th>10</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$f_1(S)$</td>
<td>96</td>
<td>94</td>
<td>92</td>
<td>90</td>
<td>88</td>
<td>86</td>
<td>84</td>
</tr>
<tr>
<td>$y_1$</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>$f_2(S)$</td>
<td>190</td>
<td>186</td>
<td>182</td>
<td>178</td>
<td>174</td>
<td>170</td>
<td></td>
</tr>
<tr>
<td>$y_2$</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$f_3(S)$</td>
<td>282</td>
<td>276</td>
<td>270</td>
<td>264</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$y_3$</td>
<td>2,4</td>
<td>2,4</td>
<td>2,4</td>
<td>2,4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\sum_{i=1}^3 y_i$</td>
<td>6,8</td>
<td>6,8</td>
<td>6,8</td>
<td>6,8</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
21. Lagrangian Multiplier, Counter Example

The previous example showed how, by varying $\lambda$, various cumulative flow rates and cost returns were generated until the proper value of $\lambda$ yielded the specified cumulative flow rate equal to 18.

There are problems in which a specified cumulative flow cannot be obtained by a $\lambda$ parameter study alone; that is to say, no matter how closely spaced the $\lambda$ values are taken, the term $\sum_{t=1}^{\lambda} y_t$ never equals the specified quantity. In these cases, the Lagrangian multiplier technique must be augmented. One supplementary method is to seek successively
the next best solution until the cumulative flow equals the specified flow. We consider here the same problem given in Section 20, except that we seek the maximum. The equations are

\[ f_N(S) = \max_{\{y_N\}} [y_N^a - S + 100 - \lambda y_N + f_{N-1}(y_N + S)], \quad N = 1, 2, \ldots \]  

(1)

with \( f_0(S) = 0 \).

The \( y_i \) terms in stages 1, 2, and 3, respectively, may assume the following values:

\[ y_1 = 2, 4, 6, 8, 10 \]  

(2)

\[ y_2 = 2, 4, 6, 8, 10 \]  

(3)

\[ y_3 = 2, 4, 6 \]  

(4)

We desire to find the maximum return, subject to the condition that \( \sum_{i=1}^{3} y_i \) equals 18.

\begin{table}[h]
\begin{center}
\begin{tabular}{llllllllll}
\hline
\multicolumn{1}{c}{\( \lambda \)} & 0 & 9 & 10 & 11 & 11 & 12 & 12 & 13 \\
\hline
\( y_1 \) & 10 & 10 & 10 & 10 & 10 & 10 & 2 & 2 \\
\( y_2 \) & 10 & 10 & 10 & 10 & 2 & 2 & 2 & 2 \\
\( y_3 \) & 6 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\
\( \sum y \) & 26 & 22 & 22 & 22 & 14 & 14 & 6 & 6 \\
f_3(0) & 514 & 292 & 264 & 248 & 248 & 234 & 234 & 228 \\
\hline
\end{tabular}
\end{center}
\end{table}

In Table 11, we list the results of the computation. In this table, we observe the discontinuity in the return function and the cumulative flow for the various \( \lambda \). This is in sharp contrast to Table 10 for the minimization, where the transition from \( \lambda \) to \( \lambda \) is marked by a smooth transition in the cumulative flow and the return function. In Table 11, we observe that for \( \lambda \) in the range \((0, 10)\) cumulative flow changed only 4 units, from 26 to 22. For \( \lambda = 11 \), the two cumulative flow rates, that earned the same return, are 22 and 14, a difference of 8 units.

If the problem had called for the return at \( \sum_{i=1}^{3} y_i = 26 \) or 22, or 14 or 6, Table 11 would give the results as readily as Table 10. The problem, however, called for \( \sum_{i=1}^{3} y_i = 18 \), and there is no \( \lambda \) which
yields this cumulative flow. In this example, it is not possible to find \( \Sigma_{i=1}^{3} y_i = 18 \) by using smaller intervals in \( \lambda \) or in \( y \). The reader is urged to test this statement by trying for himself.

The method recommended here for finding the best return for \( \Sigma_{i=1}^{3} y_i = 18 \) is to find successively the next best returns until \( \Sigma_{i=1}^{3} y_i \) does equal 18. In Table 12 for \( \lambda = 0 \), the \( k \)th best returns are ranked from the first through the 15th best, which yields the first appearance of \( \Sigma_{i=1}^{3} y_i = 18 \). The flow distribution corresponding to this is \( y_1 = 10, \ y_2 = 6, \ y_3 = 2, \) and \( f_3(0) = 430 \).

### TABLE 12

**kth Best Return**

\[
\begin{align*}
\lambda &= 0 \\

f_N(S) &= \max_{\{y_N\}} \left[ y_N^3 - S + 100 - \lambda y_N + f_{N-1}(S + y_N) \right] \\

\end{align*}
\]

<table>
<thead>
<tr>
<th>Rank</th>
<th>( y_1 )</th>
<th>( y_2 )</th>
<th>( y_3 )</th>
<th>( \sum_{i=1}^{3} y_i )</th>
<th>Return ( f_3(0) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>10</td>
<td>6</td>
<td>26</td>
<td>514</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>10</td>
<td>4</td>
<td>24</td>
<td>498</td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td>10</td>
<td>2</td>
<td>22</td>
<td>490</td>
</tr>
<tr>
<td>4</td>
<td>10</td>
<td>8</td>
<td>6</td>
<td>24</td>
<td>480</td>
</tr>
<tr>
<td>5</td>
<td>8</td>
<td>10</td>
<td>6</td>
<td>24</td>
<td>478</td>
</tr>
<tr>
<td>6</td>
<td>10</td>
<td>8</td>
<td>4</td>
<td>22</td>
<td>464</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td>10</td>
<td>4</td>
<td>22</td>
<td>462</td>
</tr>
<tr>
<td>8</td>
<td>10</td>
<td>8</td>
<td>2</td>
<td>20</td>
<td>456</td>
</tr>
<tr>
<td>9</td>
<td>10</td>
<td>6</td>
<td>6</td>
<td>22</td>
<td>454</td>
</tr>
<tr>
<td>9</td>
<td>8</td>
<td>10</td>
<td>2</td>
<td>20</td>
<td>454</td>
</tr>
<tr>
<td>10</td>
<td>6</td>
<td>10</td>
<td>6</td>
<td>22</td>
<td>450</td>
</tr>
<tr>
<td>11</td>
<td>8</td>
<td>8</td>
<td>6</td>
<td>22</td>
<td>444</td>
</tr>
<tr>
<td>12</td>
<td>10</td>
<td>6</td>
<td>4</td>
<td>20</td>
<td>438</td>
</tr>
<tr>
<td>13</td>
<td>10</td>
<td>4</td>
<td>6</td>
<td>20</td>
<td>436</td>
</tr>
<tr>
<td>14</td>
<td>6</td>
<td>10</td>
<td>4</td>
<td>20</td>
<td>434</td>
</tr>
<tr>
<td>15</td>
<td>4</td>
<td>10</td>
<td>6</td>
<td>20</td>
<td>430</td>
</tr>
<tr>
<td>15</td>
<td>10</td>
<td>6</td>
<td>2</td>
<td>18</td>
<td>430</td>
</tr>
</tbody>
</table>

In a problem as simple as this, it is tempting and reasonable to take a combinatorial approach. In this method, all possible acceptable combinations of \( y \), that sum to 18 can be tried to find that combination which yields the greatest return. For larger problems, the combinatorial approach gets out of hand. The generation of the \( k \)th best solution offers an orderly way to maximize the return and to consume exactly the required flow.
In Table 11, the desired cumulative flow rate of 18 is bracketed by the cumulative flow rates 14 and 22 in the two columns under $\lambda = 11$. For this reason, we may use $\lambda = 11$ and find the $k$th best return that yields $\sum_{i=1}^{3} y_i = 18$. Table 13 summarizes the results. Here the 5th best return produces $\sum_{i=1}^{3} y_i = 18$, and $f_3(0) = 232$.

**Table 13**

<table>
<thead>
<tr>
<th>Rank</th>
<th>$y_1$</th>
<th>$y_2$</th>
<th>$y_3$</th>
<th>$\sum_{i=1}^{3} y_i$</th>
<th>Return $f_3(0)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>10</td>
<td>2</td>
<td>22</td>
<td>248</td>
</tr>
<tr>
<td>1</td>
<td>10</td>
<td>2</td>
<td>2</td>
<td>14</td>
<td>248</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>10</td>
<td>2</td>
<td>14</td>
<td>240</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>6</td>
<td>240</td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td>8</td>
<td>2</td>
<td>20</td>
<td>236</td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td>4</td>
<td>2</td>
<td>16</td>
<td>236</td>
</tr>
<tr>
<td>4</td>
<td>10</td>
<td>10</td>
<td>4</td>
<td>22</td>
<td>234</td>
</tr>
<tr>
<td>4</td>
<td>10</td>
<td>2</td>
<td>4</td>
<td>16</td>
<td>234</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>6</td>
<td>2</td>
<td>18</td>
<td>232</td>
</tr>
</tbody>
</table>

Comparing the results of Tables 12 and 13, we note that the optimal flow distribution is identical; namely, $y_1 = 10$, $y_2 = 6$, $y_3 = 2$. The value of the return $f_3(0)$ in Table 13 is less than that in Table 12 since the $\lambda$ is larger for the Table 13 calculations. One does not know whether there will be discontinuities in the $\lambda$ and $\sum_{i=1}^{3} y_i$ terms as disclosed by Table 11, without actually calculating Table 11. If discontinuities are revealed over the region of interest, it appears reasonable to embark on the $k$th best return computation using a $\lambda$ from Table 11 that generates a total flow "close" to the desired flow. Following this procedure, in Table 13 for $\lambda = 11$, we found the solution as the 5th best return. In contrast to this in Table 12 for $\lambda = 0$, the solution appeared as the 15th best return. In other words, Table 11 may be used as sieve to isolate the best $\lambda$ to be employed in the $k$th best return calculation.

### 22. Multi-Dimensional Isoperimetric Problem

Now that we have discussed how to handle a problem with one constraint using Lagrangian multiplier, we may apply the same idea to multi-dimensional problems. In this case, each isoperimetric constraint
has its own Lagrangian multiplier determined by techniques similar to that given in Sections 19 and 20.

To illustrate, suppose we desire to treat the equation

$$ f_N(S, x, Q) = \max_{0 \leq y_N \leq x} \left[ g_N(y_N, T_N, S) + f_{N-1}(S + y_N, x - y_N, Q - y_N T_N) \right] $$

subject to

$$ \sum_{i=1}^{N} y_i = x = \text{constant} \quad (2) $$

$$ \sum_{i=1}^{N} y_i T_i \leq Q = \text{constant} \quad (3) $$

The first constraint, Eq. (2), may be considered as before to be the quantity of flow to be allocated over the $N$ stages.

The second constraint, Eq. (3), may consider to be the upper bound on the cumulative heat release.

By using Lagrangian multipliers $\lambda_1$ and $\lambda_2$, we reformulate the problem as

$$ f_N(S) = \max_{0 \leq y_N \leq x} \left[ g_N(y_N, T_N, S) - \lambda_1 y_N - \lambda_2 y_N T_N + f_{N-1}(S + y_N) \right] \quad (4) $$

While the constraint, Eq. (2), is an equality constraint and Eq. (3) is an inequality constraint, both can be handled by the Lagrangian multiplier technique. The quantities $\lambda_1$ and $\lambda_2$ are chosen arbitrarily, as before for the single constraint problem, and the return computed. By selecting a range of values for each $\lambda$, the proper values of $\lambda$ are determined numerically that satisfy both Eq. (2) and Eq. (3). Computationally, it may be better to fix one $\lambda$, perhaps $\lambda_1$, at certain values and solve the problem repeatedly for the $\lambda_2$. The process is then repeated for another value of $\lambda_1$, and for a variety of values for $\lambda_2$. The process should converge on a $\lambda_1$ and $\lambda_2$ that satisfy both constraints.

23. The kth Best Policies

There are problems in which it is desirable not only to compute the best policy, but also the hierarchy of the policies from the best to the worst. The need for this may occur when we are not too sure of our model, or when we are not too sure of the data in the model, and when we desire to have a broad prospective of our solution. In other cases in industrial practice, the effort and inconvenience of achieving the very
best just may not be worth the additional return. For example, most plant operators would object to a schedule that required a great many startups and shutdowns. We, therefore, are justified in looking for alternative, near optimal solutions. The kth best policy determination also enables us to evaluate and rank the current way of operating a process with respect to the best. The kth best policy provides a sensitivity analysis of the best policy by demonstrating the returns associated with each policy, and thereby a comparison with the optimal.

For a process where there exists a set of \( i \) possible decisions at each stage, we may set up functional equations to compute recursively the kth best policy in terms of the \((k - 1)\)th best policy.

We define

\[
f_N(S) = \text{the return from an } N\text{-stage process starting with the system in state } S \text{ and pursuing an optimal policy; } N = 1, 2, 3 ... \\
T_i(S) = \text{the } i\text{th allowable decision out of the set of decisions } \{T_i(S)\}, \text{ which results in altering the state } S \text{ in stage } i \text{ to } (S + AS_i) \text{ in stage } (i - 1); i = 1, 2, ..., m \\
g_N(T_i(S)) = \text{the return for the current stage beginning in state } S \text{ for the } i\text{th decision}
\]

We then write

\[
f_N(S) = \sum_{i} \text{Max} [g_N(T_i(S)) + f_{N-1}(S + AS_i)] \tag{1}
\]

The subscript \( i \) under the Max symbol means that the maximum is achieved over the \( i \) possible decisions in the set \( \{T_i(S)\} \).

Alternately, we may write Eq. (1) as

\[
f_N(S) = \sum_{i} \text{Max} f_{N-1}(T_i(S)), \quad i = 1, 2, ..., m \tag{2}
\]

where

\[
f_{N-1}(T_i(S)) = g_N(T_i(S)) + f_{N-1}(S + AS_i) \tag{3} \\
f_i(S) = \text{Max} g_i(T_i(S)), \quad i = 1, 2, ..., m \tag{4}
\]

To evaluate \( f_N(S) \), we must evaluate \( f_{N-1}(T_1(S)), f_{N-1}(T_2(S)), ..., f_{N-1}(T_m(S)) \) and select the largest. Equation (2) may be written to show this more explicitly:

\[
f_N(S) = \text{Max} \left[ f_{N-1}(T_1(S)), f_{N-1}(T_2(S)), ..., f_{N-1}(T_m(S)) \right] \tag{5}
\]

With these preliminaries out of the way, we now define

\[
f_N^{(k)}(S) = \text{the } k\text{th best return from the } N\text{-stage process, starting in state } S \text{ and pursuing an optimal policy; } N = 1, 2, ...
\]
The $k$th best policy yields the $k$th best return which is smaller than all the preceding returns, but at least as great as the return for the $(k + 1)$st, $(k + 2)$nd, etc., best policies.

By Eqs. (2), (4), and (5) and the definition of $f_N^{(k)}(S)$, we have, using the Principle of Optimality,

$$f_N^{(k)}(S) = \max_{i(k)} [f_{N-1}^{(1)}(T_i(S)), f_{N-1}^{(2)}(T_i(S)), \ldots, f_{N-1}^{(k)}(T_i(S))]$$  \hspace{1cm} (6)

$$f_1^{(k)}(S) = \max_{i(k)} [g_i(T_i(S))]$$ \hspace{1cm} (7)

The notation $\max_{i(k)}$ is used to convey the idea that of the $i$ possible decisions, we desire the decision that yields the $k$th best return.

Equation (7) gives the $k$th best return for a one-stage process. Because $T_i(S)$ refers to the $i$th decision and $g(T_i(S))$ refers to the return for the $i$th possible decision, we evaluate $f_1^{(k)}(S)$ by evaluating the return from each permitted policy $g_1(T_1(S)), g_1(T_2(S)), \ldots, g_1(T_m(S))$ and then rank the returns in order of decreasing values. In this manner, we find $f_1^{(1)}(S), f_1^{(2)}(S), \ldots, f_1^{(k)}(S)$.

Equation (6) is a recursion relationship between the $k$th best return in an $N$-stage process and an $(N - 1)$-stage process.

Each term on the right-hand side represents not a single value, but rather a set of return functions over the $i$ possible decisions, $T_i(S)$, $i = 1, 2, \ldots, m$. In other words, each term in Eq. (6) consists of $m$ return functions for the $i = 1, 2, \ldots, m$ possible decisions. To evaluate Eq. (6), we must write out each term of each set, evaluate them numerically, rank them, and pick out the $k$th best return. This return is by definition $f_N^{(k)}(S)$.

To be more specific, the set of return functions associated with $f_{N-1}^{(1)}(T_i(S))$ over the $i = 1, 2, 3, \ldots, m$ allowable decisions is

$$f_{N-1}^{(1)}(T_i(S)) = g_N(T_i(S)) + f_{N-1}^{(1)}(S + \Delta S_1), g_N(T_2(S)) + f_{N-1}^{(1)}(S + \Delta S_2),$$

$$\ldots, g_N(T_m(S)) + f_{N-1}^{(1)}(S + \Delta S_m)$$ \hspace{1cm} (8)

where $\Delta S_i$ represents the change in $S$ due to the $i$th allowable decision. In words, the expression $g_N(T_1(S)) + f_{N-1}^{(1)}(S + \Delta S_1)$ is the sum of the immediate return for the policy $T_1(S)$, plus the return over the $(N - 1)$ remaining stages for the first best return starting in state $(S + \Delta S_1)$. The term $g_N(T_2(S)) + f_{N-1}^{(1)}(S + \Delta S_2)$ is the sum of the immediate return for the policy $T_2(S)$ plus the return over the $(N - 1)$ remaining stages for the second best return starting in state $(S + \Delta S_2)$.
24. Numerical Example of kth Best Policy

We may, therefore, rewrite Eq. (6) as

\[
 f_{N}^{(k)}(S) = \max_{k} \left[ g_N(T_1(S) + f_{N-1}^{(1)}(S + \Delta S_1), g_N(T_2(S) + f_{N-1}^{(1)}(S + \Delta S_2), \ldots, g_N(T_m(S) + f_{N-1}^{(1)}(S + \Delta S_m), g_N(T_1(S) + f_{N-1}^{(2)}(S + \Delta S_1), g_N(T_2(S) + f_{N-1}^{(2)}(S + \Delta S_2), \ldots, g_N(T_m(S) + f_{N-1}^{(2)}(S + \Delta S_m)), \ldots, g_N(T_1(S) + f_{N-1}^{(k)}(S + \Delta S_1), g_N(T_2(S) + f_{N-1}^{(k)}(S + \Delta S_2), \ldots, g_N(T_m(S) + f_{N-1}^{(k)}(S + \Delta S_m) \right]
\]

The function \( f_{N-1}^{(1)}(T_k(S)) \) is given by the first line on the right-hand side. The function \( f_{N-1}^{(2)}(T_k(S)) \) is given by the second line on the right-hand side. The function \( f_{N-1}^{(k)}(T_k(S)) \) is given by the last line.

What Eqs. (6), (7), and (9) do, in effect, is to form all the return functions possible with the allowable decisions, then rank the returns in the order of decreasing returns, and select the kth best return. Equation (6) or (9) represents merely a clever way to evaluate combinatorially all possible combinations by using the previously determined best return functions. It is the recursion aspect of Eqs. (6), (7), and (9) that makes dynamic programming so powerful and so attractive for generating the kth best return.

24. Numerical Example of the kth Best Policy

To illustrate the utilization of Eqs. (6), (7), and (9) of Section 23, we return to the problem in Section 21:

\[
f_{N}(S) = \max_{y_i} \left[ y_N^2 - S + 100 + f_{N-1}(S + y_N) \right], \quad N = 1, 2, \ldots, f_0(S) = 0
\]

In a three-stage process, we have the following permitted policies:

In stage 1: \( y_1 = 2, 4, 6, 8, 10 \)

In stage 2: \( y_2 = 2, 4, 6, 8, 10 \)

In stage 3: \( y_3 = 2, 4, 6 \)
Referring to the definitions in the Section 23, we see the set of policies for

\begin{align*}
\text{stage 1: } & \{ T_i(S) \} = \{ y_1 = 2, 4, 6, 8, 10 \} \quad i = 1, 2, \ldots, 5 \\
\text{stage 2: } & \{ T_i(S) \} = \{ y_2 = 2, 4, 6, 8, 10 \} \quad i = 1, 2, \ldots, 5 \\
\text{stage 3: } & \{ T_i(S) \} = \{ y_3 = 2, 4, 6 \} \quad i = 1, 2, 3
\end{align*}

Observe the correspondence in the equalities

\[ g_N(T_i(S)) = g_N(S, y) = y_i^2 - S + 100 \] (6)

With this correspondence established, we now evaluate

\[ f_1^{(k)}(S) = \max_{i \in (k)} [y_i^2 - S + 100] = \max_{i \in (k)} g_1(S, y), \] (7)

Since \( y \) may assume the values of \( y_1 = 2, 4, 6, 8, 10 \), we evaluate the right-hand side of Eq. (7) for these values and rank them in decreasing order:

\begin{align*}
& f_1^{(1)}(S) = 200 - S \quad y_1^{(1)} = 10 \\
& f_1^{(2)}(S) = 164 - S \quad y_1^{(2)} = 8 \\
& f_1^{(3)}(S) = 136 - S \quad y_1^{(3)} = 6 \\
& f_1^{(4)}(S) = 116 - S \quad y_1^{(4)} = 4 \\
& f_1^{(5)}(S) = 104 - S \quad y_1^{(5)} = 2
\end{align*}

Table 14 lists \( g_1(S, y_i) \) and Table 15 lists \( f_1^{(k)}(S) \).

**TABLE 14**

<table>
<thead>
<tr>
<th>Profit for Stage ( i )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( g_1(S, y_i) = y_i^2 - S + 100 )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( y_i )</th>
<th>0</th>
<th>2</th>
<th>4</th>
<th>6</th>
<th>8</th>
<th>10</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>200</td>
<td>198</td>
<td>196</td>
<td>194</td>
<td>192</td>
<td>190</td>
<td>188</td>
</tr>
<tr>
<td>8</td>
<td>164</td>
<td>162</td>
<td>160</td>
<td>158</td>
<td>156</td>
<td>154</td>
<td>152</td>
</tr>
<tr>
<td>6</td>
<td>136</td>
<td>134</td>
<td>132</td>
<td>130</td>
<td>128</td>
<td>126</td>
<td>124</td>
</tr>
<tr>
<td>4</td>
<td>116</td>
<td>114</td>
<td>112</td>
<td>110</td>
<td>108</td>
<td>106</td>
<td>104</td>
</tr>
<tr>
<td>2</td>
<td>104</td>
<td>102</td>
<td>100</td>
<td>98</td>
<td>96</td>
<td>94</td>
<td>92</td>
</tr>
</tbody>
</table>
For a two-stage process, we set up Eq. (6) or (9) of Section 23 as

\[ f_2^{(k)}(S) = \]

\[
\max \left[ g_2(S, 10) + f_1^{(1)}(S + 10), g_2(S, 8) + f_1^{(1)}(S + 8), \ldots, g_2(S, 2) + f_1^{(1)}(S + 2), \\
g_2(S, 10) + f_1^{(2)}(S + 10), g_2(S, 8) + f_1^{(2)}(S + 8), \ldots, g_2(S, 2) + f_1^{(2)}(S + 2), \\
g_2(S, 10) + f_1^{(3)}(S + 10), g_2(S, 8) + f_1^{(3)}(S + 8), \ldots, g_2(S, 2) + f_1^{(3)}(S + 2), \\
g_2(S, 10) + f_1^{(4)}(S + 10), g_2(S, 8) + f_1^{(4)}(S + 8), \ldots, g_2(S, 2) + f_1^{(4)}(S + 2), \\
g_2(S, 10) + f_1^{(5)}(S + 10), g_2(S, 8) + f_1^{(5)}(S + 8), \ldots, g_2(S, 2) + f_1^{(5)}(S + 2) \right] 
\]

(9)

where the first line on the right-hand side corresponds to \( f_{N-1}^{(1)}(T_1(S)) \), the second to \( f_{N-1}^{(2)}(T_1(S)) \), the third to \( f_{N-1}^{(3)}(T_1(S)) \), etc. We observe that the values of \( g_2(S, 10) \), \( g_2(S, 8) \), etc., are listed in Table 14. The values for \( f_1^{(k)}(S) \) are also given in Table 15. As a consequence of this, the right-hand side of Eq. (8) may be evaluated numerically.

Table 16 lists and ranks the right-hand side of Eq. (9) for \( S = 0 \). Table 17 lists \( f_2^{(k)}(S) \) and the corresponding values of \( y_2 \) and \( y_1 \).
### TABLE 16

<table>
<thead>
<tr>
<th>kth BEST RETURN FOR TWO-STAGE PROCESS, BEGINNING IN STATE (S = 0)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(g_2(S, 10) + f_1^{11}(S + 10) = 200 + 190 = 390 = f_2^{11}(0))</td>
</tr>
<tr>
<td>(g_2(S, 8) + f_1^{11}(S + 8) = 164 + 192 = 356 = f_2^{11}(0))</td>
</tr>
<tr>
<td>(g_2(S, 6) + f_1^{11}(S + 6) = 136 + 194 = 330 = f_2^{11}(0))</td>
</tr>
<tr>
<td>(g_2(S, 4) + f_1^{11}(S + 4) = 116 + 196 = 312 = f_2^{11}(0))</td>
</tr>
<tr>
<td>(g_2(S, 2) + f_1^{11}(S + 2) = 104 + 198 = 302)</td>
</tr>
<tr>
<td>(g_2(S, 10) + f_1^{13}(S + 10) = 200 + 154 = 354 = f_2^{13}(0))</td>
</tr>
<tr>
<td>(g_2(S, 8) + f_1^{13}(S + 8) = 164 + 156 = 320 = f_2^{13}(0))</td>
</tr>
<tr>
<td>(g_2(S, 6) + f_1^{13}(S + 6) = 136 + 158 = 294)</td>
</tr>
<tr>
<td>(g_2(S, 4) + f_1^{13}(S + 4) = 116 + 160 = 276)</td>
</tr>
<tr>
<td>(g_2(S, 10) + f_1^{13}(S + 10) = 200 + 126 = 326 = f_2^{13}(0))</td>
</tr>
<tr>
<td>(g_2(S, 8) + f_1^{13}(S + 8) = 164 + 128 = 292)</td>
</tr>
<tr>
<td>(g_2(S, 6) + f_1^{13}(S + 6) = 136 + 130 = 266)</td>
</tr>
<tr>
<td>(g_2(S, 4) + f_1^{13}(S + 4) = 116 + 132 = 248)</td>
</tr>
<tr>
<td>(g_2(S, 10) + f_1^{13}(S + 10) = 200 + 106 = 306 = f_2^{13}(0))</td>
</tr>
<tr>
<td>(g_2(S, 8) + f_1^{13}(S + 8) = 164 + 108 = 272)</td>
</tr>
<tr>
<td>(g_2(S, 6) + f_1^{13}(S + 6) = 136 + 110 = 246)</td>
</tr>
<tr>
<td>(g_2(S, 4) + f_1^{13}(S + 4) = 116 + 96 = 212)</td>
</tr>
<tr>
<td>(g_2(S, 6) + f_1^{13}(S + 6) = 136 + 92 = 228)</td>
</tr>
</tbody>
</table>

In a similar manner, we determine \(f_3^{(k)}(S)\), keeping in mind that \(y_3 = 2, 4, 6\):

\[
f_3^{(k)}(S) = \max_{i \in \{4\}} \left[ g(S, 6) + f_2^{(1)}(S + 6), g(S, 4) + f_2^{(1)}(S + 4), g(S, 2) + f_2^{(1)}(S + 2), \right.
\left. \varepsilon(S, 6) + f_2^{(2)}(S + 6), g(S, 4) + f_2^{(2)}(S + 4), g(S, 2) + f_2^{(2)}(S + 2), \right.
\left. g(S, 6) + f_2^{(3)}(S + 6), g(S, 4) + f_2^{(3)}(S + 4), g(S, 2) + f_2^{(3)}(S + 2), \right.
\left. g(S, 6) + f_2^{(4)}(S + 6), g(S, 4) + f_2^{(4)}(S + 4), g(S, 2) + f_2^{(4)}(S + 2), \right.
\left. g(S, 6) + f_2^{(5)}(S + 6), g(S, 4) + f_2^{(5)}(S + 4), g(S, 2) + f_2^{(5)}(S + 2), \right.
\left. g(S, 6) + f_2^{(6)}(S + 6), g(S, 4) + f_2^{(6)}(S + 4), g(S, 2) + f_2^{(6)}(S + 2), \right.
\left. \vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \right] (10)
\]
### Table 17

**kth Best Return for Two-Stage Process, Beginning in State \( S = S_f \)**

<table>
<thead>
<tr>
<th>( f_{ik}^{(k)}(S) )</th>
<th>( S )</th>
<th>0</th>
<th>2</th>
<th>4</th>
<th>6</th>
<th>8</th>
<th>10</th>
<th>( y_2 )</th>
<th>( y_1 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( f_{3}^{11}(S) )</td>
<td></td>
<td>390</td>
<td>386</td>
<td>382</td>
<td>378</td>
<td>374</td>
<td>370</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>( f_{3}^{12}(S) )</td>
<td></td>
<td>356</td>
<td>352</td>
<td>348</td>
<td>344</td>
<td>340</td>
<td>336</td>
<td>8</td>
<td>10</td>
</tr>
<tr>
<td>( f_{3}^{13}(S) )</td>
<td></td>
<td>354</td>
<td>350</td>
<td>346</td>
<td>342</td>
<td>338</td>
<td>334</td>
<td>10</td>
<td>8</td>
</tr>
<tr>
<td>( f_{3}^{14}(S) )</td>
<td></td>
<td>330</td>
<td>326</td>
<td>322</td>
<td>318</td>
<td>314</td>
<td>310</td>
<td>6</td>
<td>10</td>
</tr>
<tr>
<td>( f_{3}^{15}(S) )</td>
<td></td>
<td>326</td>
<td>322</td>
<td>318</td>
<td>314</td>
<td>310</td>
<td>306</td>
<td>10</td>
<td>6</td>
</tr>
<tr>
<td>( f_{3}^{16}(S) )</td>
<td></td>
<td>320</td>
<td>316</td>
<td>312</td>
<td>308</td>
<td>304</td>
<td>300</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>( f_{3}^{17}(S) )</td>
<td></td>
<td>312</td>
<td>308</td>
<td>304</td>
<td>300</td>
<td>296</td>
<td>292</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>( f_{3}^{18}(S) )</td>
<td></td>
<td>306</td>
<td>302</td>
<td>298</td>
<td>294</td>
<td>290</td>
<td>286</td>
<td>10</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 18 ranks the \( f_{ik}^{(k)}(0) \) and lists the corresponding \( y_1, y_2, y_3 \).

### Table 18

**kth Best Return for Three-Stage Process, Beginning in State \( S = 0 \)**

<table>
<thead>
<tr>
<th>( f_{3}^{k}(0) )</th>
<th>( y_3 )</th>
<th>( y_2 )</th>
<th>( y_1 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( f_{3}^{11}(0) ) = 514</td>
<td>6</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>( f_{3}^{12}(0) ) = 498</td>
<td>4</td>
<td>8</td>
<td>10</td>
</tr>
<tr>
<td>( f_{3}^{13}(0) ) = 490</td>
<td>2</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>( f_{3}^{14}(0) ) = 480</td>
<td>6</td>
<td>8</td>
<td>10</td>
</tr>
<tr>
<td>( f_{3}^{15}(0) ) = 478</td>
<td>6</td>
<td>10</td>
<td>8</td>
</tr>
<tr>
<td>( f_{3}^{16}(0) ) = 464</td>
<td>4</td>
<td>8</td>
<td>10</td>
</tr>
<tr>
<td>( f_{3}^{17}(0) ) = 462</td>
<td>4</td>
<td>10</td>
<td>8</td>
</tr>
<tr>
<td>( f_{3}^{18}(0) ) = 456</td>
<td>2</td>
<td>8</td>
<td>10</td>
</tr>
<tr>
<td>( f_{3}^{19}(0) ) = 454</td>
<td>6</td>
<td>6</td>
<td>10</td>
</tr>
<tr>
<td>( f_{3}^{20}(0) ) = 454</td>
<td>2</td>
<td>10</td>
<td>8</td>
</tr>
<tr>
<td>( f_{3}^{21}(0) ) = 450</td>
<td>6</td>
<td>10</td>
<td>6</td>
</tr>
<tr>
<td>( f_{3}^{22}(0) ) = 444</td>
<td>6</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>( f_{3}^{23}(0) ) = 438</td>
<td>4</td>
<td>6</td>
<td>10</td>
</tr>
<tr>
<td>( f_{3}^{24}(0) ) = 436</td>
<td>6</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>( f_{3}^{25}(0) ) = 434</td>
<td>4</td>
<td>10</td>
<td>6</td>
</tr>
<tr>
<td>( f_{3}^{26}(0) ) = 430</td>
<td>6</td>
<td>10</td>
<td>4</td>
</tr>
<tr>
<td>( f_{3}^{27}(0) ) = 430</td>
<td>2</td>
<td>6</td>
<td>10</td>
</tr>
</tbody>
</table>
The reader, to satisfy himself that the functional equation technique truly works, should check some of the results in Tables 14–18 by the combinatorial method, as well as by dynamic programming.

25. Method of Mechanical Quadrature

Let us now discuss a technique for solving problems of higher dimensions employing the method of mechanical quadrature. Before discussing the dynamic programming problem, we briefly review the subject of mechanical quadrature.

Mechanical quadrature refers to the numerical evaluation of an integral. A typical numerical integration may be carried out using a Newton-Cotes formula such as

\[ \int_{-1}^{1} f(x) \, dx = \frac{1}{45} [7f(-1) + 32f(-\frac{1}{2}) + 12f(0) + 32f(\frac{1}{2}) + 7f(1)] + E \]  

This is a five-point formula where the function \( f(x_i) \) is known at the evenly spaced points \( x = -1, -\frac{1}{2}, 0, \frac{1}{2}, 1 \). The integral over the range \((-1, 1)\) can be approximately evaluated numerically from a knowledge of these five points. Under appropriate assumptions, the error \( E \) in the value of the integral by Eq. (1) does not exceed

\[ \frac{1}{15,120} \left| \frac{df^6(\epsilon)}{dx^6} \right| \]

where \( \epsilon \) is a point within \((-1, 1)\).

Another way to evaluate \( \int_{-1}^{1} f(x) \, dx \) is by a formulation that chooses the spacing along the abscissa in some optimal way (which is generally not evenly spaced). "Optimal way" means here requiring fewer points, yielding higher accuracy formulas, and employing orthogonal polynomials with interesting and useful properties.

A typical formula of this type is the Legendre-Gauss three-point formula:

\[ \int_{-1}^{1} f(x) \, dx = \frac{1}{9} \left[ 5f \left( \frac{\sqrt{15}}{5} \right) + 8f(0) + 5f \left( \frac{\sqrt{15}}{5} \right) \right] + E \]  

Here, the error \( E \) does not exceed

\[ \frac{1}{15,750} \left| \frac{df^6(\eta)}{dx^6} \right| \]

where \( \eta \) is a point within \((-1, 1)\).
In this formula, Eq. (2), the points used are \( x = \pm \sqrt{15}/5 \) and 0. The points chosen are the zeros of the Legendre polynomial \( P_3(x) \); that is, the values of \( x \) that make \( P_3(x) = 0 \), where \( P_3(x) = \frac{1}{2}(5x^3 - 3x) \). A comparison between the five-point Newton-Cotes formula and the three-point Legendre-Gauss is fair since the error terms are approximately the same.†

The Newton-Cotes formula is based on an interpolation polynomial where the interpolation polynomial and \( f(x) \) agree at \((n + 1)\) equally spaced points. Least squares fitting is not employed.

The Legendre-Gauss formula is derived by least squares fitting using orthogonal polynomials [41]. The orthogonal polynomial (in this case, Legendre polynomials), as with all orthogonal polynomials, requires fewer terms in the normal equations of the least squares fit than is the case in fitting with other functions. As a consequence, fewer terms appear in the Legendre-Gauss formula.

In the case of the Legendre-Gauss quadrature, it can be shown that

\[
\int_{-1}^{1} f(x) \, dx = \sum_{j=1}^{s} H_j f(x_j) + E
\]

where \( x_j \) is the \( j \)th zero of the Legendre polynomial \( P_s(x) \) of degree \( s \). The error \( E \) is zero if \( f(x) \) is a polynomial of degree \((2s - 1)\) or less. Here \( s \) is the upper bound on the summation and \( H_j \) is defined by

\[
H_j = \frac{2(1 - x_j^2)}{(s + 1)^2 [P_{s+1}(x_j)]^2}, \quad j = 1, 2, \ldots, s
\]

Use of Eq. (3) requires that the function \( f(x_j) \) be evaluated at the points \( x_j \). The \( x_j \) points \((x_1, x_2, \ldots, x_s)\) are known since they are the

† Legendre polynomials can be determined by the recursion relationship

\[
P_{s+1}(x) = \frac{2s + 1}{s + 1} x P_s(x) - \frac{s}{s + 1} P_{s-1}(x)
\]

where \( P_s(x) \) = the Legendre polynomial of degree \( s \).

The first few Legendre polynomials are

\[
\begin{align*}
P_0(x) &= 1 \\
P_1(x) &= x \\
P_2(x) &= \frac{1}{2} (3x^2 - 1) \\
P_3(x) &= \frac{1}{2} (5x^3 - 3x) \\
P_4(x) &= \frac{1}{8} (35x^4 - 30x^2 + 3)
\end{align*}
\]

Legendre polynomials are orthogonal in the region \((-1, 1)\)

\[
\int_{-1}^{1} P_s(x) P_r(x) \, dx = \begin{cases} 
\frac{2}{2s + 1}, & s = r \\
0, & s \neq r
\end{cases}
\]

The normalized polynomials are \((2/(2s + 1))^{1/2} P_s(x)\).
zeros of the Legendre polynomial $P_s(x)$. Corresponding to each of the $x_j$ points is the weight $H_j$ in Eq. (4).

To demonstrate the use of Eqs. (3) and (4) let us consider the case where $s = 3$. The Legendre polynomial is

$$P_3(x) = \frac{1}{2} (5x^3 - 3x)$$

The zeros of $P_3(x)$ are found from

$$P_3(x) = 0 = \frac{1}{2} (5x^3 - 3x)$$

They are $x_1 = -\sqrt{15}/5$, $x_2 = 0$, $x_3 = \sqrt{15}/5$.

Since $P_{s+1}(x) = P_4(x) = \frac{1}{8}(35x^4 - 30x^2 + 3)$ and since the zeros of $P_3(x)$ are known, the values of $H_j$ in Eq. (4) are $H_1 = 5/9$, $H_2 = 8/9$, $H_3 = 5/9$.

Substituting these $H_j$ and $x_j$ into Eq. (3), we have the approximate formula

$$\int_{-1}^{1} f(x) \, dx = \frac{5}{9} f\left(-\frac{\sqrt{15}}{5}\right) + \frac{8}{9} f(0) + \frac{5}{9} f\left(\frac{\sqrt{15}}{5}\right)$$

which is Eq. (2).

In summary, to use the Legendre-Gauss formula, Eq. (3), one first decides upon the degree of the approximating Legendre polynomial. This fixes $s$. With $P_s(x)$ decided upon, the zeros of $P_s(x)$ may be found in tables or from the formulas given in the footnote on page 215. The function $f(x)$ is evaluated at each of the zeros of $P_s(x)$ to give $f(x_j)$ and $H_j$ is found from Eq. (4) or from tables. In fact, the $H_j$ terms may be determined, once and for all, once the degree of the Legendre polynomial is fixed. If $f(x)$ is a polynomial of degree $R$, and if the degree of the approximating Legendre polynomial is $s$, then for $R \leq 2s - 1$, the error term in Eq. (3) vanishes because the derivative $df^{2s}/dx^{2s}$ does not exist. In other words, by mechanical quadrature the integral of a polynomial of degree $R$ may be evaluated without error by Legendre polynomial of lower degree, namely, $s \geq (R + 1)/2$.

26. Mechanical Quadrature and Dynamic Programming

To return to the dynamic programming problem now, let us consider a sequence of functions:

$$f_1(c) = \max_v g(c, v)$$

$$f_N(c) = \max_v [g(c, v) + f_{N-1}(h(c, v))]$$

where $c$ exists over $(-1, 1)$. 
This problem might be solved by setting up a grid in \( c \) and computing \( f_N(c) \) starting with \( f_1(c) \) as previously described.

Another approach is to describe \( f_N(c) \) in terms of an orthogonal polynomial expansion such as

\[
f_N(c) = \sum_{k=0}^{R} a_{k,N} P_k(c)
\]  

(3)

The function \( f_N(c) \) is a polynomial of degree \( R \), and \( P_k(c) \) is the normalized Legendre polynomial of degree \( k \), described above. The \( a_{k,N} \) are coefficients which are normally determined by multiplying both sides by \( P_i(c) \), \( i = 0, 1, \ldots, R \) and integrating over the range \((-1, 1)\). From the orthonormal properties of these polynomials

\[
\int_{-1}^{1} P_i(c) P_j(c) \, dc = \begin{cases} 
0, & i \neq j \\
1, & i = j
\end{cases}
\]  

(4)

we have

\[
a_{k,N} = \int_{-1}^{1} f_N(c) P_k(c) \, dc
\]  

(5)

At this point, we seem to be chasing ourselves in a circle, since \( f_N(c) \) by Eq. (3) depends on \( a_{k,N} \), and in Eq. (5), \( a_{k,N} \) depends on \( f_N(c) \). Which comes first? If we know \( f_N(c) \), there is no point in finding \( a_{k,N} \) since the problem is solved. However, we cannot find \( a_{k,N} \) without \( f_N(c) \).

To extricate ourselves from this quandry, let us use mechanical quadrature:

\[
\int_{-1}^{1} f_N(c) P_s(c) \, dc = \sum_{j=1}^{s} a_j f_N(c_j) P_k(c_j), \quad k = 0, 1, 2, \ldots, R
\]  

(6)

where the \( c_j \)'s are the zeros of the Legendre polynomial of degree \( s \) and \( f_N(c) \) is considered to be a polynomial of degree \( R \).

Comparing Eqs. (5) and (6), we see that

\[
a_{k,N} = \sum_{j=1}^{s} a_j f_N(c_j) P_k(c_j), \quad k = 0, 1, 2, \ldots, R
\]  

(7)

Let us now note the correspondence between Eq. (6) of this section and Eq. (3), Section 25:

\[
f(x) = f_N(c) P_k(c)
\]

\[
f(x_j) = f_N(c_j) P_k(c_j)
\]

\[
H_j = a_j
\]  

(8)
To get started in the calculation using this method of mechanical quadrature requires first that the $f_N(c_j)$ be known. Referring specifically to Eqs. (1) and (2), we must evaluate first $f_1(c_j)$ for $c_1$, $c_2$, ..., $c_s$, the zeros of the Legendre polynomial of degree $s$. Selecting a grid of $c_1$, $c_2$, ..., $c_s$, we evaluate $f_1(c_j)$ as usual, for a one-stage process by the proper choice of $v$ in Eq. (1). With the knowledge of $f_1(c_j)$, we next desire to evaluate $f_1(c)$ for any value $c$ in the range $(-1, 1)$. This is accomplished by evaluating $a_i = H_i$ by Eq. (4), Section 25 and $P_k(c_j)$ by tables or the Legendre polynomial recursion equations. Then $a_{k,1}$ is found for $k = 0, 1, 2, ..., R$ by Eq. (7). Once the $a_{k,1}$ are known, then by Eq. (3) we can evaluate $f_1(c)$ for any $c$ within $(-1, 1)$. Note that $f_1(c)$ is curve fitted from the values of the $f_1(c_j)$ terms. The function $f_1(c)$ is not found directly by the proper choice of $v$ in Eq. (1).

For the two-stage process, we evaluate first $f_2(c_j)$ by means of Eq. (2). The choice of the value of $v$ that maximizes the right-hand side of Eq. (2) transforms the state of the system from $c_j$ into $c_j + \Delta c$. Since the new state of the system ($c_j + \Delta c$) will, in general, not be a zero of the Legendre polynomial of degree $s$, the term $f_1(c_j + \Delta c)$ can be evaluated from the coefficients $a_{k,1}$ which were developed from the one-stage process. Once the $f_2(c_j)$ terms are evaluated, the coefficients $a_{k,2}$, $k = 0, 1, 2, ..., R$ can be generated from Eq. (7). From $a_{k,2}$ and Eq. (3), $f_2(c)$ is found for general values of $c$. Again, $f_2(c)$ is curve fitted from $f_2(c_j)$. In this manner, we develop $f_N(c)$. It should be recognized that the Legendre polynomials $P_k(c_j)$ and the coefficients $a_j$ can be calculated independently of $f_N(c_j)$.

For the two dimensional case we have

$$f_N(c_1, c_2) = \sum_{k,l=0}^{R} a_{k,l,N} P_k(c_1) P_l(c_2) \tag{9}$$

where

$$a_{k,l,N} = \int_{-1}^{1} \int_{-1}^{1} f_N(c_1, c_2) P_k(c_1) P_l(c_2) \, dc_1 \, dc_2 \tag{10}$$

and

$$a_{k,l,N} = \sum_{j=1}^{s} \sum_{r=1}^{s} a_j a_r f_N(c_{1j}, c_{2r}) P_k(c_{1j}) P_l(c_{2r}) \tag{11}$$

The term $f_N(c_1, c_2)$ is a polynomial of degree $R$, the $c_{1j}$ and $c_{2r}$ are the roots of the Legendre polynomial of degree $s$, and $a_j$ and $a_r$ are computed as before by Eq. (8) and by Eq. (4) of Section 25. The Legendre polynomials $P_k(c_{1j})$ and $P_l(c_{2r})$ and the coefficients $a_j$ and $a_r$ are evaluated independently of $f_N(c_{1j}, c_{2r})$. 
In a similar manner problems of higher dimensionality are handled by mechanical quadrature.

The significance of the method of mechanical quadrature to dynamic programming is its utility for high-dimensional problems. As we have pointed out many times, the storing of the conventional multi-dimensional grid in computer memory becomes prohibitive due to computer memory size limitations. On the other hand, the storing of the coefficients $a_{k,N}$ requires considerably less room due to the fact that space may be saved by using a low degree approximating Legendre polynomial, without sacrificing accuracy.

To illustrate the relative space requirements in the one-dimensional case, a set of $(R + 1)$ constants $a_{k,N}$ must be stored at each stage. For the one-dimensional case with the conventional grid, using a grid spacing of $\Delta$ units, $1/\Delta$ units must be stored. In this case, the method of mechanical quadrature seldom has any real advantage. For a two-dimensional case $(R + 1)^2$ coefficients must be stored compared to $1/\Delta^2$ units to be stored for the two-dimensional grid. For three dimensions, the number of coefficients is $(R + 1)^3$, while the grid requires $1/\Delta^3$ terms.

The memory required at each stage for conventional grid technique versus the method of mechanical quadrature is listed in Table 19.

<table>
<thead>
<tr>
<th>Dimension $N$</th>
<th>Conventional Grid memory requirements $(1/\Delta)^N$</th>
<th>Mechanical quadrature $R = 5$</th>
<th>Mechanical quadrature $R = 10$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$10^8$</td>
<td>6</td>
<td>11</td>
</tr>
<tr>
<td>2</td>
<td>$10^4$</td>
<td>36</td>
<td>121</td>
</tr>
<tr>
<td>3</td>
<td>$10^6$</td>
<td>216</td>
<td>1331</td>
</tr>
<tr>
<td>4</td>
<td>$10^8$</td>
<td>1296</td>
<td>14,641</td>
</tr>
<tr>
<td>5</td>
<td>$10^{10}$</td>
<td>7776</td>
<td>161,051</td>
</tr>
</tbody>
</table>

In summary, we may say that the method of mechanical quadrature in dynamic programming is a compromise in which an exchange is made of smaller memory requirements for longer computer time.
27. Reduction in Dimensionality by Linearization. Terminal Control

We consider in this section the reduction in dimensionality of a terminal control problem in the calculus of variations.

In particular, we consider the problem of determining the maximum of $J(y)$:

$$J(y) = H[x_1(T), x_2(T), ..., x_k(T)], \quad N > k$$

subject to the relations

$$\frac{dx_i}{dt} = G_i(x, y), \quad i = 1, 2, ..., N$$

$$x_i(0) = c_i, \quad i = 1, 2, ..., N$$

$$m_i \leq y_i(t) \leq M_i, \quad 0 \leq t \leq T$$

$$\int_0^T L(y_1, y_2, ..., y_N) \, dt \leq L_1$$

Here $x$ and $y$ are, respectively, vectors with components $[x_1(t), x_2(t), ..., x_N(t)]$ and $[y_1(t), y_2(t), ..., y_N(t)]$. In this development, we exploit the fact that there are only $k$ variables in the criterion function, where $N > k$, and the fact that $G_i(x, y)$, a nonlinear function, in general may be approximated by a linear function. The important property of linear equations that is used is the additivity of the solution of the homogeneous linear equation and the particular solution.

We will solve first the terminal control problem in the normal way, using the technique of Chapter 4. The standard way of solving this problem is to express the state of the system in terms of the initial values of the $x_i$, namely, the $c_i$ terms, which are $N$ in number, and the duration of the process.

We redefine $J(y)$ as

$$J(y) = H[x_1(T), x_2(T), ..., x_k(T)] + \lambda \int_0^T L(y_1, y_2, ..., y_N) \, dt$$

where $\lambda$ is a Lagrangian multiplier, and set

$$f(c_1, c_2, ..., c_N, T) = \max_{y_i} J(y)$$

At the end of the process, we have

$$f(c_1, c_2, ..., c_N, 0) = H[x_1(T), x_2(T), ..., x_k(T)]$$

$$= H[c_1, c_2, ..., c_k]$$
Following the usual procedure of breaking the time period \((0, T)\) into two time periods \((0, S)\) and \((S, T)\), we have for small \(S\)

\[
f(c_1, c_2, \ldots, c_N, T) = \max_{v_i} \left[ \lambda L(v_1, v_2, \ldots, v_N) S + f(c_1 + G_1(c_1, v_1) S, c_2 + G_2(c_2, v_2) S, \ldots, c_N + G_N(c_N, v_N) S, T - S) \right] (9)
\]

where \(v_i = y_i(0)\).

Equation (9), together with the terminal condition, Eq. (8), represents one form of solution.

We may go further and expand the right-hand side of Eq. (9) by a Taylor's series to obtain

\[
\frac{\partial f(c_1, c_2, \ldots, c_N, T)}{\partial T} = \max_{v_i} \left[ \lambda L(v_1, v_2, \ldots, v_N) + \sum_{i=1}^{N} \frac{\partial f}{\partial c_i} G_i \right] (10)
\]

This equation, plus the terminal condition, Eq. (8), establishes the equations of the terminal control problem.

Before discussing the reduction in dimensionality technique, we review some important properties of linear systems.

## 28. Some Properties of Linear Systems

Let us consider the set of differential equations expressed in vector form

\[
\frac{dx}{dt} = Ax + By, \quad x(0) = c (1)
\]

When the matrix \(A\) is independent of time, the solution of Eq. (1) is

\[
x = e^{At}c + \int_0^t e^{A(t-s)}By(s) \, ds (2)
\]

The important thing to note is that

\[
x = e^{At}c (3)
\]

is the solution to the homogeneous equation

\[
\frac{dx}{dt} = Ax, \quad x(0) = I (4)
\]

and it is completely independent of \(B\) and \(y(s)\).
The complete solution is, therefore, composed of the solution to the homogeneous equation, plus the particular solution due to $By$. The analogy to scalar differential equations is complete.

It follows, then, that at $t = T$

$$x(T) = e^{AT}c + \int_0^T e^{A(T-s)}By(s) \, ds \quad (5)$$

If we let

$$b = e^{AT}c \quad (6)$$

and

$$R(s) = e^{AT-s}B \quad (7)$$

then Eq. (5) becomes

$$x(T) = b + \int_0^T R(s) y(s) \, ds \quad (8)$$

Here $b$ and $R(s)$ are fixed if $T$ is fixed.

For each component of $x(T)$, we have

$$x_i(T) = b_i + \int_0^T \sum_{j=1}^N r_{ij}(s) y_j(s) \, ds, \quad i = 1, 2, ..., N \quad (9)$$

Again, we point out that $b_i$ is completely independent of the control vector $y$.

### 29. Terminal Control for Linearized Equations

If now we return to Eq. (2) of Section 27, and assume for the moment that $G_t(x, y)$ can be approximated by a linear expression such as Eq. (1) of Section 28, then we can use Eq. (9) of Section 28 to describe $x_i(T)$.

We now define, as in Section 27, $J(y)$ as

$$J(y) = H[x_1(T), x_2(T), ..., x_k(T)] + \lambda \int_0^T L(y_1, y_2, ..., y_N) \, dt \quad (1)$$

Let us now substitute Eq. (9) of Section 28 into Eq. (1):

$$J(y) = H \left[ b_1 + \int_0^T \sum_{j=1}^N r_{1j}(s) y_j(s) \, ds, b_2 + \int_0^T \sum_{j=1}^N r_{2j}(s) y_j(s) \, ds, ..., b_k + \int_0^T \sum_{j=1}^N r_{kj}(s) y_j(s) \, ds \right] + \lambda \int_0^T L(y_1, y_2, ..., y_N) \, dt \quad (2)$$
This is an equation in \( y_i \) only. The \( b_i \) terms are independent of the \( y_i \). By the substitution of Eq. (9) of Section 28, we have eliminated \( x_i \) entirely.

We now define

\[
f(b_1, b_2, ..., b_k, T) = \max_{u_i} J(y) \tag{3}
\]

where the state variables \( b_1, b_2, ..., b_k \) are the solution of the homogeneous linear Eq. (2) of Section 27, after it has been linearized. The duration of the process is, again, expressed by \( T \). We use here only \( k \) of the \( b_i \) terms, which corresponds to the \( k \) terminal conditions in the criterion function \( H[x_1(T), x_2(T), ..., x_k(T)] \).

As usual, we break the time interval \((0, T)\) into \((0, S)\) and \((S, T)\):

\[
f(b_1, b_2, ..., b_k, T) = \\
\max_{u_i} \left\{ H \left[ b_1 + \int_0^S \sum_{j=1}^N r_{ij}(s) y_j(s) \, ds + \int_S^T \sum_{j=1}^N r_{ij}(s) y_j(s) \, ds, \right. \right. \\
\left. b_2 + \int_0^S \sum_{j=1}^N r_{2j}(s) y_j(s) \, ds + \int_S^T \sum_{j=1}^N r_{2j}(s) y_j(s) \, ds, \right. \\
\left. \left. \ldots \ldots \ldots \ldots \ldots \right] \\
\left. b_k + \int_0^S \sum_{j=1}^N r_{kj}(s) y_j(s) \, ds + \int_S^T \sum_{j=1}^N r_{kj}(s) y_j(s) \, ds \right] \right. \right. \\
+ \lambda \int_0^S L(y_1, y_2, ..., y_N) \, dt + \lambda \int_S^T L(y_1, y_2, ..., y_N) \, dt \right\} \tag{4}
\]

We note that over the interval \((0, S)\) for a choice of \( y_i \) that the original value of \( b_i \) is increased to

\[
b_i + \int_0^S \sum_{j=1}^N r_{ij}(s) y_j(s) \, ds
\]

Since the \( b_i \) and \( r_{ij}(s) \) are known, for a choice of \( y_i(s) \), the new value of \( b_i \) can be computed. Let us call \( b'_i \) the new value of \( b_i \) at time \( S \).

\[
b'_i = b_i + \int_0^S \sum_{j=1}^N r_{ij}(s) y_j(s) \, ds \tag{5}
\]
Now, if we refer to the definition of \( f(b_1, b_2, ..., b_k, T) \) in Eq. (3), we may write

\[
f(b_1', b_2', ..., b_k', T - S) = \max_{y} \left\{ H \left[ b_1' + \int_0^T \sum_{j=1}^N r_{ij}(s) y_j(s) \, ds, \right.ight.
\]
\[
\left. b_2' + \int_0^T \sum_{j=1}^N r_{2j}(s) y_j(s) \, ds, ..., \right.
\]
\[
\left. b_k' + \int_0^T \sum_{j=1}^N r_{kj}(s) y_j(s) \, ds \right\}
\]

(6)

Substituting Eqs. (5) and (6) into Eq. (4), we obtain

\[
f(b_1, b_2, ..., b_k, T) = \max_{v} \left\{ \lambda \int_0^T L(y_1, y_2, ..., y_N) \, dt + f(b_1', b_2', ..., b_k', T - S) \right\}
\]

(7)

For small \( S \), we have, by Eqs. (5) and (7),

\[
f(b_1, b_2, ..., b_k, T) = \max_{v} \left\{ \lambda L(v_1, v_2, ..., v_N) S \right.
\]
\[
+ f \left[ b_1 + S \sum_{j=1}^N r_{1j}(s) v_j(s), b_2 + S \sum_{j=1}^N r_{2j}(s) v_j(s), ..., b_k + S \sum_{j=1}^N r_{kj}(s) v_j(s), T - S \right] \right\}
\]

(8)

where \( v_i = y_i(0) \).

The terminal boundary condition yields

\[
f(b_1, b_2, ..., b_k, 0) = H[x_1(T), x_2(T), ..., x_k(T)]
\]

(9)

or

\[
f(b_1, b_2, ..., b_k, 0) = H(b_1, b_2, ..., b_k)
\]

(10)

Equation (9) or Eq. (10) states that a process of zero duration length, measured from the final state, is at the final state.

We may solve Eq. (8) by working back from Eq. (9). For example, if we are one time increment from the end of the process, Eq. (8) appears as

\[
f(b_1, b_2, ..., b_k, (T - 1 \cdot S)) = \max_{v} \{ \lambda(L(v_1, v_2, ..., v_N) S + f(b_1', b_2', ..., b_k', 0) \}
\]

(11)

where the \( b_i' \) are defined by Eq. (5).
Since the function \( f(b'_1, b'_2, ..., b'_k, 0) \) is known by Eq. (9), we may evaluate \( f(b'_1, b'_2, ..., b'_k, (T - 1 \cdot S)) \) by a suitable choice of \( v \). In a similar manner, we develop \( f(b'_1, b'_2, ..., b'_k, (T - 2S)) \) in terms of \( f(b'_1, b'_2, ..., b'_k, (T - 1 \cdot S)) \).

If \( k = 1 \) or 2, the proposed method is immediately applicable. Since the control criterion may specify only one or two terminal conditions for the \( x_i(T) \) out of many \( x_i \) terms, this method offers an interesting and practical way to reduce dimensionality.

It is also possible for us to expand Eq. (8) in a Taylor series to obtain

\[
\frac{\partial f(b_1, b_2, ..., b_k)}{\partial T} = \max_v \left\{ \lambda L(v_1, v_2, ..., v_N) + \sum_{i=1}^{k} \frac{\partial f}{\partial b_i} \sum_{j=1}^{N} r_{ij}(s) v_j \right\}
\]

The numerical evaluation of this equation with Eq. (9) as an initial value condition is another possibility for solution.

We may go one step further and form the set of equations consisting of Eqs. (13), (14), and (10):

\[
\frac{\partial}{\partial v_i} \left[ \lambda L(v_1, v_2, ..., v_N) + \sum_{i=1}^{k} \frac{\partial f}{\partial b_i} \sum_{j=1}^{N} r_{ij}(s) v_j \right] = 0, \quad i = 1, 2, ..., N \quad (13)
\]

\[
\frac{\partial f(b_1, b_2, ..., b_k)}{\partial T} = \lambda L(v_1, v_2, ..., v_N) + \sum_{i=1}^{k} \frac{\partial f}{\partial b_i} \sum_{j=1}^{N} r_{ij}(s) v_j \quad (14)
\]

\[
f(b_1, b_2, ..., b_k, 0) = H(b_1, b_2, ..., b_k) \quad (10)
\]

Equations (13) and (14) are equivalent to Eq. (12), and Eq. (10) supplies the terminal condition.

At this point, let us clearly distinguish the difference in point of view between the normal way of solving the terminal problem and the linearization technique. In the normal method, we define the state of the system in terms of the initial values of \( x_i \), namely, the \( c_i \) terms, which are \( N \) in number, and the duration of the process. The transition of the system from state to state is carried out by using Eq. (2) of Section 27, so that the new value of \( c_i \) equals \( c_i + G_i(c, v) \cdot S \). The development of the nonlinear partial differential equation, Eq. (10) of Section 27, is generated using the differential Eq. (2) of Section 27 and a limiting process. The terminal condition of maximizing \( H[x_1(T), x_2(T), ..., x_k(T)] \) is handled as a boundary condition.

In the linearized point of view, we do not use the differential equation, Eq. (2) of Section 27, to describe the transition from state to state. In place of it, we use the integrated form of the linearized version of Eq (2) of Section 27, namely, Eq. (2) of Section 28, to describe the transition from
state to state. In particular, the transition is given by Eq. (5) of this section. As a result of this linearization, the solution to the homogeneous part of the linearized Eq. (2) of Section 27, namely, the $b_i$ terms, is completely independent of the forcing terms $y_i$. It is precisely for this reason that $k$ state variables, rather than $N$ state variables can be used in the linear system to characterize the state of the system. We note by Eq. (6) of Section 28 that the initial condition $x_i(0) = c_i$ is absorbed in the definition of $b_i$. In developing the equations of the process, we have two integral terms with which to deal. The first integral term is the forcing function integral term, namely,

$$\int_0^T \sum_{j=1}^{N} r_{ij}(s) y_j(s) \, ds$$

and the second is the constraint integral

$$\int_0^T L(y_1, y_2, ..., y_N) \, dt$$

30. Successive Linear Approximations

The reduction in dimensionality of the terminal control problem hinges entirely on the linearization of the functions $G_i(x, y)$. To approximate suitably the $G_i(x, y)$, we employ a method of successive approximations.

Let us refer to the problem of maximizing

$$J(y) = H[x_1(T), x_2(T), ..., x_k(T)]$$

subject to

$$\frac{dx_i}{dt} = G_i(x, y), \quad i = 1, 2, ..., N$$

$$x_i(0) = c_i, \quad i = 1, 2, ..., N$$

$$m_i \leq y_i(t) \leq M_i$$

$$\int_0^T L(y_1, y_2, ..., y_N) \, dt \leq L_1$$

We approximate in policy space by choosing arbitrarily an initial guess for each component of the $y$ vector over time:

$$y^{(0)}(t) = [y_1^{(0)}(t), y_2^{(0)}(t), ..., y_N^{(0)}(t)]$$

where Eq. (6) satisfies Eqs. (2)–(5).
Substituting Eq. (6) into Eq. (2), we solve for the vector \( x^{(0)}(t) \):

\[
x^{(0)}(t) = [x_1^{(0)}(t), x_2^{(0)}(t), ..., x_N^{(0)}(t)]
\]

(7)

Now armed with \( y^{(0)}(t) \) and \( x^{(0)}(t) \), we expand in a Taylor's series around the quantity \( G_i(x^{(0)}, y^{(0)}) \) of Eq. (2) as follows:

\[
\frac{dx_i}{dt} = G_i(x^{(0)}, y^{(0)}) + \sum_{j=1}^{N} (x_j - x_j^{(0)}) \frac{\partial G_i(x^{(0)}, y^{(0)})}{\partial x_j} + \sum_{j=1}^{N} (y_j - y_j^{(0)}) \frac{\partial G_i(x^{(0)}, y^{(0)})}{\partial y_j}, \quad i = 1, 2, ..., N
\]

(8)

This is a linear equation in \( x_j \) and \( y_j \) and the set of these equations may be cast into the form of Eq. (1) of Section 28.

With this linear approximation for Eq. (2), the approach and results of Section 29 are applicable. As a consequence of Section 29, we develop the set of \( y_i \) that optimize. We call this vector \( y^{(1)}(t) \).

Once again we use \( y^{(1)}(t) \) to find \( x^{(1)}(t) \) from Eq. (2). The next linear approximation is given by

\[
\frac{dx_i}{dt} = G_i(x^{(1)}, y^{(1)}) + \sum_{j=1}^{N} (x_j - x_j^{(1)}) \frac{\partial G_i(x^{(1)}, y^{(1)})}{\partial x_j} + \sum_{j=1}^{N} (y_j - y_j^{(1)}) \frac{\partial G_i(x^{(1)}, y^{(1)})}{\partial y_j}
\]

Referring once again to Section 29 where we use this linear approximation, we generate the next optimizing vector \( y^{(2)}(t) \). Substituting \( y^{(2)}(t) \) into Eq. (2), we solve for \( x^{(2)}(t) \).

Proceeding in this manner, we generate a sequence of vectors \( y^{(0)} \), \( x^{(0)} \), \( y^{(1)} \), \( x^{(1)} \), \( y^{(2)} \), \( x^{(2)} \), etc. When \( y^{(N+1)} \approx y^{(N)} \), the process terminates.

This technique of successive approximation may involve a great deal of computation. For each approximation in policy space, the linearized equations are solved by the techniques of Section 29. Since the solution for each approximation may be lengthy, one may anticipate a sizeable computing problem. This, however, may be a small price to pay for a problem that might not be solved otherwise.

### 31. State of System Specified by Distribution Function

In the problems discussed up to this point, the state of the system has been specified by specific variables such as \( x \) in the function \( f_N(x) \)
or $x$ and $y$ in the function $f_N(x, y)$. In the formulations, we have been able to specify how $x$ and $y$ change from stage $N$ to stage $(N-1)$. As a consequence of this, we have been able to set up recursion relationships, and in particular, computable relationships.

There is an extremely large class of problems where the state of the system is not specified by a small number of specific variables as above but by a distribution function such as $\varphi(x)$. The defining expressions are given in terms of $f_N(\varphi(x))$. The quantity $f_N(\varphi(x))$ is a function of a function and is, therefore, called a functional. In these problems, an infinite number of variables are required to describe the state of the system.

Typical problems that are described by a distribution function such as $\varphi(x)$ rather than by specific variables such as $x$, are compressible flow problems. Included in these are the minimization of compressor power costs in a gas pipeline. In this problem, the state of the system is specified by the gas density distribution as a function of the pipeline length.

Problems characterized by a distribution function have not been successfully attacked in general by dynamic programming. They represent a challenging area for mathematical exploration and discovery. The situation where the state of the system is described by a distribution function is difficult to handle due to the problem of specifying the relationships from state to state as the process proceeds from stage to stage. When the state variables are discrete and bounded, say $0 \leq S \leq 1$, and monotonic, both the range and the sequence of the state variables are known. In the case where the state is specified by a distribution function, one does not know, in general, how the shape of the function will change from stage to stage.

To get around this difficult problem of specifying the state of the system by a distribution, one would like to somehow characterize the distribution function in a simple manner and still retain all or essentially all of the information contained in the state distribution function itself.

The problem, therefore, reduces itself to finding parameters (few in number) that characterize the state functions. The concept of expressing the behavior of a distribution function by certain parameters is well known in statistics; for example, the mean and the variance of the Gaussian distribution specify completely the Gaussian distribution. If the state of the system were characterized by a Gaussian distribution, we would not have to deal with a state specified by $S(\varphi, \sigma)$.

Computationally, this

$$S = \varphi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \exp \left[ - \left( \frac{x - \mu}{\sigma} \right)^2 \right] dx$$

but rather by a state specified by $S = S(\mu, \sigma)$. Computationally, this
amounts to working with the state of a system specified by the discrete variables $\mu$ and $\sigma$. This also means that the state does not have to be identified in the computer by many values of $x$ and $\varphi(x)$.

If the state of the system were specified by a linear function,

$$S = \varphi(x) = mx + b$$

we would deal only with $S = S(m, b)$. We can specify completely the linear distribution by only its slope, $m$, and intercept, $b$.

In many problems, the distribution function is not known beforehand explicitly. It is known, however, that the state of the system is characterized by some distribution function, which is obtained only by solving a set of differential or partial differential equations. To obtain the characteristic parameters of the distribution, if they exist, it is suggested that the set of differential or partial differential equations be solved for a wide range of conditions. This information opens up several possible paths:

1. A study of the distribution function of the set of equations hopefully will reveal some characterizing parameters.
2. Perhaps various transformations may be attempted on this data in the search for the characterizing parameters.
3. Another possibility is curve fitting the results and using the coefficients in the curve fitted equations as the characterizing parameters.

To summarize, we may say that when the state of the system is characterized by a distribution function, we must reduce this to a few variables. Thus, we reduce the $f_N(\varphi(x))$, a functional to, say, $f_N(x_1, x_2)$ a function.

32. Dynamic Programming and Linear Programming

Linear programming deals with optimization problems where there is a linear objective function $F$ to be minimized or maximized

$$F = \sum_{j=1}^{N} c_j x_j, \quad j = 1, 2, ..., N$$

subject to a set of linear constraints

$$\sum_{j=1}^{N} a_{ij} x_j \leq b_i, \quad i = 1, 2, ..., M; \quad j = 1, 2, ..., N$$
where

\[ N \geq M, \quad x_j \geq 0, \quad a_{ij} \geq 0 \] (3)

Problems of this type for large numbers of variables and large numbers of constraints can be solved using the simplex algorithm of Dantzig. In these problems, the \( x_j \) are allowed to assume any of a continuous range of values. Certain problems involving discrete values for the \( x_j \) have also been solved. In particular, problems where \( x_j = 0, \) or \( 1 \) have been solved. Recently, with the introduction of integer linear programming by Gomory, many more problems dealing with the \( x_j \) as integers have been solved. The principle advantage of solving these problems by the simplex method or its variations is that relatively large problems can be handled by an efficient algorithm.

What can dynamic programming offer in this realm? For problems of small dimension, dynamic programming offers the following:

(1) It provides an alternate way to solve the linear programming problem.

(2) For problems which start out as linear programming problems, but must be made more realistic by the introduction of nonlinearities, dynamic programming provides a general approach useful for linear and nonlinear problems.

(3) Dynamic programming is not hampered by restrictions that the \( x_i \) terms must be continuous, discrete, or integers. In fact, dynamic programming can very well handle mixed linear programming problems where some of the \( x_i \)'s are continuous and others are integers.

(4) Dynamic programming emphasizes the structure of the solution.

(5) It also provides a uniform way to handle both deterministic and stochastic linear programming problems.

### 33. Maximization over a Discrete Set

In order to set the ground work for the dynamic programming approach to linear optimization problems, let us first discuss a more general problem of maximization over discrete sets.

We consider a problem here with a nonlinear objective function and nonlinear constraints where the \( x_i \) take on discrete values in the set \( S_i. \)

Let

\[ F(x) = \sum_{i=1}^{N} F_i(x_i) \] (1)
subject to the constraints

\[ \sum_{j=1}^{N} G_{ij}(x_j) \leq b_i, \quad i = 1, 2, ..., k \]  

(2)

\[ G_{ij}(x_j) \geq 0 \]  

(3)

\[ x_i \in S_i \]  

(4)

Equation (4) states the values of the \( x_i \) must be in the set of values specified by the set \( S_i \).

We may or may not require that

\[ x_i \geq 0 \]  

(5)

and the \( F_i(x_i) \) and \( G_{ij}(x_j) \) may be nonlinear functions.

We desire to maximize \( F(x) \) over the choice of the \( x_i \).

Let us define

\[ f_N(b_1, b_2, ..., b_k) = \max_{x_i} \sum_{i=1}^{N} F_i(x_i) \]  

(6)

where the \( b_i \) terms are the right-hand terms of Eq. (2). For the one-stage process, we have

\[ f_1(b_1, b_2, ..., b_k) = \max_{x_i} F_1(x_i) \]  

(7)

where

\[ G_{11}(x_1) \leq b_1, \quad G_{21}(x_1) \leq b_2, ..., G_{k1}(x_1) \leq b_k \]  

(8)

For stage 1, the \( x_i \) takes on any of the permitted values specified by Eq. (4).

For the \( N \)-stage process, using the Principle of Optimality, we have

\[ f_N(b_1, b_2, ..., b_k) = \max_{x_N} [F_N(x_N) + f_{N-1}(b_1 - G_{1N}(x_N), b_2 - G_{2N}(x_N), ..., b_k - G_{kN}(x_N))] \]  

(9)

where

\[ G_{1N}(x_N) \leq b_1, \quad G_{2N}(x_N) \leq b_2, ..., G_{kN}(x_N) \leq b_k \]  

(10)

The values that \( x_N \) take are specified by the set \( S_N \).

34. Extension to Linear Programming

If now we specialize the functions in Section 33, so that \( F_i(x_i) \) and \( G_{ij}(x_j) \) are linear, we have a form of the linear programming problem.
Let

\[ F_i(x_i) = c_i x_i \]  
(1)

\[ G_{ij}(x_j) = a_{ij} x_j; \quad a_{ij} \geq 0 \]  
(2)

Then we may restate the problem in Section 33: maximize \( \sum_{i=1}^{N} F_i(x_i) \):

\[ \sum_{i=1}^{N} F_i(x_i) = c_1 x_1 + c_2 x_2 + \ldots + c_N x_N \]  
(3)

subject to

\[ \sum_{j=1}^{N} G_{ij}(x_j) = \sum_{j=1}^{N} a_{ij} x_j \leq b_i, \quad i = 1, 2, \ldots, k \]  
(4)

\[ x_i \geq 0, \quad a_{ij} \geq 0 \]  
(5)

We define now

\[ f_1(b_1, b_2, \ldots, b_k) = \text{Max} \sum_{i=1}^{N} c_i x_i \]  
(6)

For the one-stage process, we write

\[ f_1(b_1, b_2, \ldots, b_k) = \text{Max} c_1 x_1 \]  
(7)

subject to

\[ a_{11} x_1 \leq b_1, \quad a_{21} x_1 \leq b_2, \ldots, a_{k1} x_1 \leq b_k \]  
(8)

For the \( N \)-stage process, using the Principle of Optimality, we write

\[ f_N(b_1, b_2, \ldots, b_k) = \text{Max} \left[ c_N x_N + f_{N-1}(b_1 - a_{1N} x_N, b_2 - a_{2N} x_N, \ldots, b_k - a_{kN} x_N) \right] \]  
(9)

subject to

\[ a_{1N} x_N \leq b_1, \quad a_{2N} x_N \leq b_2, \ldots, a_{kN} x_N \leq b_k \]  
(10)

Equations (7) and (9) provide a computable algorithm. The only limitation in their use is the problem of dimensionality in the state variables. Provided the number of state variables is small, 1, 2, and 3, the dynamic programming approach is practical.

The capacity of dynamic programming to deal with the \( x_i \) variables whether they are continuous, discrete, integers, or combinations of these is assured due to the numerical nature of the algorithm, Eqs. (7) and (9). To solve the problem we set up a search procedure that restricts itself to certain discrete or integer values as specified in the problem
statement or searches "continuously" for the optimizing \( x_i \) at each stage. The search procedure may be aided by noting that the set of constraints Eq. (10) and the condition Eq. (5) may be written as

\[
0 \leq x_N \leq \min \left[ \frac{b_1}{d_{1N}}, \frac{b_2}{d_{2N}}, \ldots, \frac{b_k}{d_{kN}} \right]
\]  

(11)

If \( x_N \) is restricted to certain discrete values or certain integer values, we recognize that the value of \( x_N \) from Eq. (11) probably will not be one of the set of discrete or integer values acceptable. In this case, we must choose the discrete value or integer value that comes closest to satisfying the constraint Eq. (11).

35. Reduction of Dimensionality in Linear Programming

As in all forms of dynamic programming, the reduction of dimensionality is extremely important to generate numerical results. We will discuss here two ways to reduce the problem dimensionality. The first is to take advantage of the homogeneity of the linear equations. The second way is to use, as before, Lagrangian multipliers.

Let us consider again the problem of maximizing

\[
\sum_{i=1}^{N} F_i(x_i) = c_1x_1 + c_2x_2 + \ldots + c_Nx_N
\]

subject to the two equations

\[
\sum_{j=1}^{N} a_{ij}x_j \leq b_1
\]

(2)

\[
\sum_{j=1}^{N} a_{2j}x_j \leq b_2
\]

(3)

\[
a_{ij} \geq 0, \quad c_i \geq 0
\]

(4)

From the development in Section 34 we have the functional equations:

\[
f_N(b_1, b_2) = \max_{x_N} [c_Nx_N + f_{N-1}(b_1 - a_{1N}x_N, b_2 - a_{2N}x_N)]
\]

(5)

\[
f_i(b_1, b_2) = \max_{x_1} (c_ix_1)
\]

(6)

where \( x_N \) is bound by

\[
0 \leq x_N \leq \left( \frac{b_1}{a_{1N}}, \frac{b_2}{a_{2N}} \right)
\]  

(7)
By the homogeneity of Eqs. (1)-(3), we observe that

\[ f_N(b_1, b_2) = b_1 f_N \left( 1, \frac{b_2}{b_1} \right) = b_2 f_N \left( \frac{b_1}{b_2}, 1 \right) \] (8)

Using this fact, we write Eq. (5) as

\[ f_N(b_1, b_2) = b_1 f_N \left( 1, \frac{b_2}{b_1} \right) = \max_{x_N} \left[ c_N x_N + (b_1 - a_{1N} x_N) f_{N-1} \left( 1, \frac{b_2}{b_1} - a_{2N} x_N \right) \right] \] (9)

\[ f_N \left( 1, \frac{b_2}{b_1} \right) = \max_{x_N} \left[ c_N x_N \frac{b_1}{b_2} + \left( 1 - \frac{a_{1N} x_N}{b_1} \right) f_{N-1} \left( 1, \frac{b_2}{b_1} - \frac{a_{2N} x_N}{b_2} \right) \right] \] (10)

Similarly, we obtain

\[ f_N \left( \frac{b_1}{b_2}, 1 \right) = \max_{x_N} \left[ c_N x_N \frac{b_2}{b_1} + \left( 1 - \frac{a_{2N} x_N}{b_2} \right) f_{N-1} \left( \frac{b_1}{b_2}, 1 - \frac{a_{1N} x_N}{b_1} \right) \right] \] (11)

By taking advantage of the homogeneity, we have reduced the original problem in two state variables \( b_1 \) and \( b_2 \) into a problem of one state variable, namely, the ratio \( \frac{b_2}{b_1} \) in Eq. (10) or \( \frac{b_1}{b_2} \) in Eq. (11).

By redefining the variables, we may see the reduction to one state variable a little better.

Let

\[ \frac{b_2}{b_1} = r \] (12)

\[ \frac{x_N}{b_1} = z \] (13)

We define

\[ u_N(r) = f_N \left( 1, \frac{b_2}{b_1} \right) = f_N(1, r) \] (14)

Equation (9) appears as

\[ u_N(r) = \max_{z} \left[ c_N z + (1 - a_{1N} z) f_{N-1} \left( 1, \frac{r - a_{2N} z}{1 - a_{1N} z} \right) \right] \] (15)

where

\[ 0 \leq z \leq \min \left[ \frac{1}{a_{1N}}, \frac{r}{a_{2N}} \right] \] (16)

As defined above in Eq. (12), \( 0 \leq r \leq \infty \).

To keep \( r \) bounded, we compute \( f_N(r, 1) \) and \( f_N(1, r) \) where \( 0 \leq r \leq 1 \).
When
\[(1 - a_{1N}x) \geq r - a_{2N}x\] (17)
we use Eq. (10).

When
\[(1 - a_{1N}x) < r - a_{2N}x\] (18)
we use Eq. (11).

In some problems, the expressions comparable to Eq. (5) are

Here the original state variable \(b_1\) and \(b_2\), in stage \(N\), are increased to \(b_1 + a_{1N}x_N, b_2 + a_{2N}x_N\) in stage \((N - 1)\). This problem of enlarging the state space with each stage of the process makes for computational difficulties. Because the grid of discrete state variables must be made larger and larger to carry out the numerical computation, this problem is referred to as the problem of the expanding grid (see Section 5).

The exploitation of the homogeneity aspect of the linear system and the device given in Eqs. (17) and (18) to bound \(r\), effectively eliminate the expanding grid problem.

36. Lagrangian Multipliers in Linear Programming

We consider the same problem as in the previous section except we have the additional constraint

\[\sum_{j=1}^{N} a_{3j} x_j \leq b_3\] (1)

We form a new objective function to maximize \(P_i(x_i)\)

\[P_i(x_i) = \sum_{i=1}^{N} c_i x_i - \lambda \sum_{j=1}^{N} a_{3j} x_j\] (2)

where \(\lambda\) is a Lagrangian multiplier which is arbitrarily chosen and held fixed over the \(N\) stages.

We define

\[f_N(b_1, b_2) = \max_{x_N} [c_N x_N - \lambda a_{2N} x_N + f_{N-1}(b_1 - a_{1N} x_N, b_2 - a_{2N} x_N)]\] (3)

We then set up the functional equations

\[f_N(b_1, b_2) = \max_{x_N} [c_N x_N - \lambda a_{2N} x_N + f_{N-1}(b_1 - a_{1N} x_N, b_2 - a_{2N} x_N)]\] (4)
5. COMPUTATIONAL ASPECTS

where \( x_N \) is subject to

\[
0 \leq x_N \leq \min \left( \frac{b_1}{a_{1N}}, \frac{b_2}{a_{2N}} \right) \tag{5}
\]

Equation (4) is solved for the \( \lambda \) chosen and the term \( \sum_{j=1}^{N} a_{3j}x_j \) is computed which is required to satisfy condition Eq. (1). The problem is solved over and over again for a range of \( \lambda \) values until the maximum is found over the \( \lambda \)'s. The Lagrangian multiplier technique is discussed in Section 19.

The problem can be reduced to functions of one variable by introducing a second Lagrangian multiplier. The objective function \( P_i(x_i) \) is now expressed as

\[
P_i(x_i) = \sum_{i=1}^{N} c_{ix_i} - \lambda_1 \sum_{j=1}^{N} a_{3j}x_j - \lambda_2 \sum_{j=1}^{N} a_{4j}x_j \tag{5}
\]

The functional equation is

\[
f_N(b_1) = \max_{x_N} \left[ c_N x_N - \lambda_1 a_{3N}x_N^2 - \lambda_2 a_{4N}x_N + f_{N-1}(b_1 - a_{1N}x_N) \right] \tag{6}
\]

subject to

\[
0 \leq x_N \leq \frac{b_1}{a_{1N}} \tag{7}
\]

Another way to reduce the problem to one variable is to combine the technique of Section 35 with the Lagrangian multiplier.

Starting with Eq. (3) of this section, we may express the functional equation as

\[
f_N(b_1, b_2) = b_1 f_N \left( 1, \frac{b_2}{b_1} \right) \tag{8}
\]

\[
= \max_{x_N} \left[ c_N x_N - \lambda a_{3N}x_N + \left( b_1 - a_{1N}x_N \right) f_{N-1} \left( 1, \frac{b_2}{b_1} - \frac{a_{2N}x_N}{b_1} \right) \right] \tag{9}
\]

Let, as before,

\[
\frac{b_2}{b_1} = r \tag{10}
\]

\[
x_N = \frac{b_1}{a_{1N}} \tag{11}
\]

\[
f_N(1, r) = \max_{x} \left[ (c_N - \lambda a_{3}) x + (1 - a_{1N}x) f_{N-1} \left( 1, \frac{r - a_{2N}x}{1 - a_{1N}x} \right) \right] \tag{12}
\]
REFERENCES


44. Roberts, S. M., Higher dimensionality and the control of a batch reaction. To be published.


PROBLEMS

1. For the reactor system discussed in Section 10, let

\[ g_1 = y_1 \exp \left( -\frac{T_1}{T_{\text{Max}}} \right), \quad g_2 = y_2 T_2 \exp \left( -\frac{T_2}{T_{\text{Max}}} \right) \]

\[ g_3 = \sqrt{y_3} \left( \frac{T_3}{T_{\text{Max}}} \right)^2 \exp \left( -\frac{y_3 T_3}{T_{\text{Max}}} \right) \]

\[ \sum_{i=1}^{3} y_i = 6; \quad 500 \leq T_i \leq 1000 \]

Maximize the profit \( P = \sum_{i=1}^{3} g_i \) by

(a) differential calculus
(b) dynamic programming.

2. For the problem in Section 11, solve for the initial approximation

(a) \( y_0 = x \)

(b) \( y_0 = \ln x \)

3. Consider a three-stage parallel reactor system where the profit for each stage is given by \( g_i \)

\[ g_1 = y_1 \exp \left( -\frac{T_1}{T_{\text{Max}}} \right), \quad g_2 = y_2 T_2 \exp \left( -\frac{T_2}{T_{\text{Max}}} \right) \]

\[ g_3 = \sqrt{y_3} \left( \frac{T_3}{T_{\text{Max}}} \right)^2 \exp \left( -\frac{y_3 T_3}{T_{\text{Max}}} \right) \]

If the system is constrained as follows:

(a) \( \sum_{i=1}^{3} y_i = 6, \quad y_i = 0, 1, 2, 3, 4, 5, 6 \)

(b) \( 500 \leq T_i \leq 1000, \quad T_i = 500, 600, 700, 800, 900, 1000 \)

(c) \( \sum_{i=1}^{3} y_i T_i < 5000 \)

determine the maximum profit.

Is there more than one set of values for the Lagrangian multipliers \( \lambda_1 \) and \( \lambda_2 \) that satisfy this problem?

4. In Problem 3, determine the return and policies for the first, second, third, fourth, and fifth best policy.

What is the least best policy and the associated return?
5. In Section 13, evaluate the policy and return function for an initial policy
   (a) \( y_0 = x \)
   (b) \( y_0 = \frac{1}{2} x \)
   Compare with the results of Section 13.

6. In Section 14, evaluate the policy and return function for
   (a) \( f_0(x) = \sqrt{x} \)
   (b) \( f_0(x) = x^2 \)

7. If \( g(y) = \sqrt{y} \) and \( h(x - y) = (x - y)^2 \), choose the initial policy so that

   \[
   \frac{g(y)}{(1 - a)y} = \frac{h(x - y)}{(1 - b)x}
   \]

   and evaluate

   \[
   f(x) = \text{Max} [g(y) + h(x - y) + f(ay + b(x - y))]
   \]

8. Apply the results of Section 18 to Problem 7.

9. Describe the numerical computation of

   \[
   \frac{\partial f(c, T)}{\partial T} = \text{Max}_v \left[ F(c, v) + G(c, v) \frac{\partial f(c, T)}{\partial c} \right]
   \]

   if

   \[
   \frac{\partial f(c, T)}{\partial T} \approx \frac{f(c, T + \Delta) - f(c, T)}{\Delta}
   \]

   and if

   \[
   \frac{\partial f(c, T)}{\partial c} \approx \frac{f(c + \delta, T) - f(c - \delta, T)}{2\delta}
   \]

10. Consider the grid bound by the vertical lines \( A \) and \( B \).

Let \( n(i, j; k, l) \) = the number on the line segment joining \((i, j)\) to \((k, l)\).
Show that the minimum from vertical line $A$ to vertical line $B$ is given by

$$f(i, j) = \min \{ n(i, j; i + 1, j) + f(i + 1, j), n(i, j; i + 1, j - 1) + f(i + 1, j - 1) \}$$

where

$$f(i, j) = \text{the minimum sum starting at the point } (i, j) \text{ and using an optimal path to the line } B.$$  


11. Given the following grid where the $n(i, j; k, l)$ numbers are shown:

![Grid Image]

show that the best path starts from $(0, 2)$ with cost of 12. Show also the path through the grid from line $A$ to line $B$.

Numbers on the grid below show cumulative time from each node to $B$.

![Grid Image]

12. Given the following grid bound by the vertical lines $A$ and $B$:

![Grid Image]

What is the least path from line $A$ to line $B$?
13. If in Problem 11

\[ p_k = \text{probability of moving horizontally for decision } k \]
\[ 1 - p_k = \text{probability of moving diagonally for decision } k \]

show that the minimum expected sum is given by

\[
f(i, j) = \min_k \{ p_k [n(i, j; i + 1, j) + f(i + 1, j)] + (1 - p_k) [n(i, j; i + 1, j - 1) + f(i + 1, j - 1)] \}
\]

If \( p_1 = \frac{1}{2} \) and \( p_2 = \frac{3}{4} \), show the optimal starting point is \((0, 1)\) and the expected cost is \( \frac{471}{32} \).

14. Given the array \( [a_{ij}] \)

\[
\begin{array}{cccc}
8 & 3 & 4 & 4 \\
9 & 4 & 7 & 3 \\
1 & 6 & 5 & 2 \\
2 & 5 & 1 & 6 \\
\end{array}
\]

find the path from the lower left-hand corner \( a_{41} \) to the upper right-hand digit \( a_{14} \) so that the sum of the numbers is the least. The path must be taken by moving only horizontally or vertically one step at a time. \textit{Answer:} 22. Find the longest path.

(Cartaino, T. F., and Dreyfus, S. E., Application of dynamic programming to the airplane minimum time to climb problem. \textit{Aeronaut. Eng. Rev.} \textbf{16}, 74–77 (June 1957).)

15. Given the set of equations

\[
\begin{align*}
\begin{array}{c}
a_{11}x_1 + a_{12}x_2 = b_1 \\
a_{21}x_1 + a_{22}x_2 + a_{23}x_3 = b_2 \\
\vdots \\
a_{N, N-1}x_{N-1} + a_{N, N}x_N = b_N
\end{array}
\end{align*}
\]

The matrix \( A \) of these equations is considered symmetric and positive definite. Define

\[
f_k(Z) = \min_{x_k} \left[ \sum_{i,j=1}^{k} a_{ij}x_i x_j - 2 \sum_{i=1}^{k-1} b_i x_i - 2Z x_k \right]
\]
Making use of the fact that \( a_{ij} = a_{ji} \), show that

\[
\begin{align*}
 f_k(Z) &= \min_{x_k} \left[ a_{kk}x_k^2 - 2Zx_k + \min_{x_i} \left\{ \sum_{i,j=1}^{k-1} a_{ij}x_i x_j \right\} \right. \\
 &\quad \left. - 2 \sum_{i=1}^{k-2} b_i x_i - 2(b_{k-1} - a_{k-1,k} x_k) x_{k-1} \right] \\
 f_1(Z) &= \min_{x_1} \left[ a_{11}x_1^2 - 2Zx_1 \right] = - Z/|a_{11}|
\end{align*}
\]

(Bellman, R., On some applications of dynamic programming to matrix theory. *Illinois J. Math.*, 1, No. 2, 297–301 (1957).)

16. Taking advantage of the fact that each member of the sequence \( f_k(Z) \) is quadratic in \( Z \) in Problem 15, let us set

\[
f_k(Z) = u_k + v_k Z + w_k Z^2
\]

where \( u_k, v_k, w_k \) are independent of \( Z \). Show that

\[
\begin{align*}
 (a) \quad f_k(Z) &= \min_{x_k} \left[ a_{kk}x_k^2 - 2Zx_k + u_{k-1} + v_{k-1}(b_{k-1} - a_{k-1,k} x_k) \right. \\
 &\quad \left. + w_{k-1}(b_{k-1} - a_{k-1,k} x_k)^2 \right] \\
 (b) \quad x_k &= \frac{Z + \frac{1}{2} v_{k-1} a_{k-1,k} + a_{k-1,k} w_{k-1} b_{k-1}}{a_{kk} + a_{k-1,k}^2 w_{k-1}} \\
 (c) \quad w_k &= - \frac{1}{a_{kk} + a_{k-1,k}^2 w_{k-1}} \\
 v_k &= - \frac{-v_{k-1} a_{k-1,k}}{a_{kk} + a_{k-1,k}^2 w_{k-1}} \\
 w_k &= u_{k-1} + v_{k-1} b_{k-1} + w_{k-1} b_{k-1}^2 + \frac{w_{k-1} b_{k-1}^2}{a_{kk} + a_{k-1,k}^2 w_{k-1}}
\end{align*}
\]

17. Consider the slightly intertwined system of linear equation

\[
\begin{align*}
 a_{11}x_1 + a_{12}x_2 + a_{13}x_3 &= c_1 \\
 a_{21}x_1 + a_{22}x_2 + a_{23}x_3 &= c_2 \\
 a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + b_1 x_4 &= c_3
\end{align*}
\]
The matrix is symmetric and positive definite. Let us define the matrices

\[ A_k = (a_{i+j-3k, j+i-3k}), \quad i, j = 1, 2, 3 \]

and vectors

\[ x^k = (x_{3k-2}, x_{3k-1}, x_{3k}), \quad c^k = (c_{3k-2}, c_{3k-1}, c_{3k}) \]

Using the following notation for inner product

\[ (x, Ax) = \sum_{i,j=1}^{k} a_{ij}x_i x_j \]

we may write a set of equations whose solution is equivalent to the slightly intertwined set, namely,

\[ (x^1, A_1x^1) + (x^2, A_2x^2) + \ldots + (x^N, A_Nx^N) \]

\[ - 2(c^1, x^1) - 2(c^2, x^2) - \ldots - 2(c^N, x^N) \]

\[ + 2b_1x_3 x_4 + 2b_2x_6 x_7 + \ldots + 2b_{N-1}x_{3N-6}x_{3N-5} \]

Let us define for \( N = 1, 2, \ldots, \) and \(-\infty < Z < \infty\)

\[ f_N(Z) = \min_{x_{3i}} \left[ \sum_{i=1}^{N} (x^i, A_i x^i) - 2 \sum_{i=1}^{N} (c^i, x^i) + 2 \sum_{i=1}^{N-1} b_{i+3} x_{3i} + 2Z x_{3N} \right] \]

Show that

\[ f_N(Z) = \min_{z_{3N-2}^{N-1} \neq 2 \neq 3N-2} \left[ (x^N, A_N x^N) + 2Z x_{3N} - 2(c^N, x^N) + f_{N-1}(b_{N-1} x_{3N-2}) \right] \]

18. Show that \( f_N(Z) \) in Problem 17 can be reduced to

\[ f_N(Z) = \min_{y} \left[ g_N(Z, y) + f_{N-1}(b_{N-1}y) \right] \]

where

\[ y = x_{3N-2} \]

\[ g_N(Z, y) = \min_{z_{3N} \neq 3N-1} \left[ (x^N, A_N x^N) + 2Z x_{3N} - 2(c^N, x^N) \right] \]
What is the advantage of this computational form? Discuss the feasibility of using the form

\[ f_N(Z) = U_N + 2V_N Z + W_N Z^2 \]

where \( U_N, V_N, \) and \( W_N \) are independent of \( Z \).

19. In a plant during the \( i \)th time period the product produced is designated as \( x_i \), while the demand for product is designated as \( r_i \). The demand may be satisfied by consuming the production \( x_i \) and/or by withdrawing from inventory \( Z \). We desire to minimize the cost of production while still meeting the demand required. Let

\[ \phi_i(x_i) = \text{the cost function for the } i \text{th period. It may be linear or nonlinear} \]

\[ f_k(Z) = \min \sum_{i=k}^{N} \phi_i(x_i) = \text{the minimum cumulative cost of satisfying the demand } r_k, r_{k+1}, \ldots, r_N \text{ beginning with the inventory } Z \text{ at stage } k; \text{ time is counted forward so } k = N \text{ refers to the } N \text{th or last stage} \]

The permissible region is denoted by:

\[ x_k \geq r_k - Z; \quad x_k + x_{k+1} \geq r_{k+1} - Z; \quad x_k + x_{k+1} + \ldots + x_N \geq r_N - Z \]

where \( x_i \geq 0 \).

Show that for stage \( N \) only and that

\[ f_N(Z) = \min_{x_N \geq 0} \left[ \phi_N(x_N) = \phi_N(\max(0, r_N - Z)) \right] \]

for stage \( N \) only and that

\[ f_k(Z) = \min_{x_k \geq 0} \left[ \phi_k(x_k) + f_{k+1}(x_k) \right], \quad k = 1, 2, \ldots, N - 1 \]

(Bellman, R. [1].)

20. Given

\[ \phi_k(x_k) = x_k^2 \]

and \( r_k \) as follows:

\[
\begin{array}{cccccccccc}
  k : & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\
  r_k : & 1 & 2 & 3 & 4 & 5 & 4 & 3 & 2 & 1 & 0
\end{array}
\]

Evaluate \( f_k(Z) \) in Problem 19 for \( Z = 0, 1, 2, 3, \ldots \).
Given
(b) \( \phi_k(x_k) = Kx_k \)
and \( r_k \) as follows:

\[
\begin{array}{cccccccccc}
  k : & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\
  r_k : & 1 & 2 & 3 & 4 & 5 & 4 & 3 & 2 & 1 & 0 \\
\end{array}
\]

Evaluate \( f_k(Z) \) by dynamic programming. Evaluate by linear programming. Evaluate by the use of the Langrangian multiplier.

21. Suppose in the previous examples the relative size of the change in production \( x_i \) from time stage to time stage is restricted by

\[ |x_{i+1} - x_i| \leq d_{i+1} \]

where \( d_{i+1} \) are specified. Justify the following definition and functional equations:

\[
\begin{align*}
  f_k(Z, x_{k-1}) &= \min \sum_{i=k}^{N} \phi_i(x_i) \\
  f_k(Z, x_{k-1}) &= \min_{x_k \geq r_k - Z} \left[ \phi_k(x_k) + f_{k+1}(Z + x_k, x_k) \right], \quad k = 2, 3, ..., N \\
  f_N(Z, x_{N-1}) &= \min [\phi_N(x_k)] \\
\end{align*}
\]

The permissible region is defined by

\[
\begin{align*}
  x_i &\geq 0, \quad i = k, k + 1, ..., N \\
  x_k &\geq r_k - Z \\
  x_k + x_{k+1} &\geq r_{k+1} - Z \\
  x_k + x_{k+1} + ... + x_N &\geq r_N - Z \\
  |x_{r+1} - x_r| &\leq d_{r+1}, \quad r = k - 1, k, ..., N - 1 \\
\end{align*}
\]

If the ratio of the production in period \( (i + 1) \) to the production in period \( i \) is restricted by

\[
\frac{x_{i+1}}{x_i} < a_{i+1}
\]

where the \( a_{i+1} \) are known constants, how is the problem formulated?

Suppose the demand \( r_i \) cannot be met without violating at one or more stages \( |x_{i+1} - x_i| \leq d_{i+1} \). Discuss how dynamic programming can be used to determine the nature of the solution vis-à-vis relaxing the constraint \( (x_{i+1} - x_i) \leq d_{i+1} \), restricting \( (r_{i+1} - r_i) \), or examining the ranges of \( (r_{i+1} - r_i) \) that are satisfactory.
22. Given the linear expression

\[ L(x) = \sum_{k=0}^{N-1} [a, x(k)] \]

where \( x(k) \) satisfies a vector expression \( x(k+1) = Ax(k) + By(k) \), \( x(0) = c \). The independent variable \( y(k) \) is constrained by

\[ Cy(k) \leq Dx(k), \quad y(k) \geq 0 \]

The \( a, A, B, C, \) and \( D \) are known constants. Show that the \( L(x) \) may be maximized by the functional equations

\[
\begin{align*}
 f_N(c) &= \text{Max} L(x) \\
 f_N(c) &= \text{Max} \left\{ [a, x(0)] + \sum_{k=1}^{N-1} (a, x(k)) \right\} \\
 f_N(c) &= \text{Max} \left\{ (a, c) + f_{N-1}(x(1)) \right\} \\
 f_N(c) &= \text{Max} \left\{ (a, c) + f_{N-1}(Ac + By) \right\}
\end{align*}
\]

(Bellman, R. [14].)

23. Given

\[
 f(x) \text{ Max}_{0 \leq y \leq x} \left[ re^{-e/y} + e^{-d/x-y} + f(ay + b(x - y)) \right]
\]

and the values

\[
\begin{align*}
 a &= 0.8 & d &= 15 \\
 b &= 0.9 & r &= 1 \\
 c &= 10
\end{align*}
\]

Develop the return \( f(x) \) and the policy \( y(x) \) for an infinite stage process starting with

\[
\begin{align*}
 f_1(x) &= \text{Max}_{0 \leq y \leq x} \left[ re^{-e/y} + e^{-d/x-y} \right] \\
 f_N(x) &= \text{Max}_{0 \leq y \leq x} \left[ re^{-e/y} + e^{-d/x-y} + f_1(ay + b(x - y)) \right]
\end{align*}
\]

(Bellman, R., Computational problems in the theory of dynamic programming, The RAND Corp. P-423 (Aug. 6, 1953)

24. Suppose we desire to stock a warehouse with the maximum value of merchandise. The warehouse capacity is limited by the volumetric capacity of the warehouse, as well as total weight to be carried by the floor. If there are \( N \) total items with the \( i \)th item of weight \( W_i \), volumetric size \( S_i \), and value \( V_i \), how many items \( X_i \) of type \( i = 1, 2, \ldots, N \) may be stored?
The volumetric capacity of the warehouse is

\[ S \geq \sum_{i=1}^{N} S_i X_i \]

The weight limitation of the warehouse is given by

\[ W \geq \sum_{i=1}^{N} W_i X_i \]

The value of the total stock is \( \sum_{i=1}^{N} V_i X_i \) where \( X_i \) is the number of items of type \( i \) to be stored, an integer. Let us define

\[ f_R(W', S') = \text{the maximum return from a warehouse of weight capacity } W' \text{ and volumetric capacity } S' \text{ for the first } R \text{ type of items} \]

Show that return and policy are determined by

\[ f_R(W', S') = \max_{X_R} [V_R X_R + f_{R-1}(W' - X_R, S' - X_R S_R)] \]

where

\[ X_R = 0, 1, 2, ..., \min\left\{ \frac{W'}{W_R}, \frac{S'}{S_R} \right\} \]

Are these alternate formulations acceptable?

(a) \[ f_R(W') = \max_{X_R} [V_R X_R - \lambda S_R X_R + f_{R-1}(W' - X_R W_R)] \]

(b) \[ f_R(S') = \max_{X_R} [V_R X_R - \lambda W_R X_R + f_{R-1}(S' - X_R S_R)] \]

(Dreyfus, S. E. [38].)

25. Maximize \( Z = x_1 + x_2 + 3x_3 \) subject to

\[ 3x_1 + 2x_2 + x_3 \leq 3, \quad 2x_1 + x_2 + 2x_3 \leq 2 \]

by

(a) simplex algorithm

(b) differential calculus and Lagrangian multiplier

(c) dynamic programming

Show that

\[ x_1 = 0, \quad x_2 = \frac{1}{3}, \quad x_3 = \frac{1}{3}, \quad Z = \frac{7}{3} \]
26. Maximize \( z = 2x_1 + 4x_2 + x_3 + x_4 \) subject to
\[
x_1 + 3x_2 + x_4 \leq 4, \quad 2x_1 + x_2 \leq 3, \quad x_2 + 4x_3 + x_4 \leq 3
\]
Show that the optimal result is
\[
x_1 = 1, \quad x_2 = 1, \quad x_3 = \frac{1}{2}, \quad x_4 = 0, \quad z = 6.5
\]
(Amundson, N. R., Aris, R., Kalman, R. E., and Lapidus, L., A.I.Ch.E. Special Lecture Series, p. 66–69 (December 3, 1960).)

27. Show that the following three representations are equivalent:

(a) \( f_N(x, S, Q) = \max_{0 \leq y_N \leq x} [g_N(y_N, S, T_N) + f_{N-1}(x - y_N, S + y_N, Q - a_N T_N)] \)
subject to
\[
\sum_{i=1}^{N} y_i = x, \quad \sum_{i=1}^{N} a_i T_i \leq Q
\]

(b) \( f_N(S, Q) = \max_{0 \leq y_N \leq x} [g_N(y_N, S, T_N) - \lambda_1 y_N + f_{N-1}(S + y_N, Q - a_N T_N)] \)
subject to
\[
\sum_{i=1}^{N} y_i = x, \quad \sum_{i=1}^{N} a_i T_i \leq Q
\]

(c) \( f_N(S) = \max_{0 \leq y_N \leq x} [g_N(y_N, S) - \lambda_1 y_N - \lambda_2 a_N T_N + f_{N-1}(S + y_N)] \)
subject to
\[
\sum_{i=1}^{N} y_i = x, \quad \sum_{i=1}^{N} a_i T_i \leq Q
\]
Discuss the merits of each representation.
6 Control Problems

1. Introduction

Control problems appear in a variety of processes including biological, economic, engineering, chemical, and physical. A general theme runs through all control problems. First, there is an equation or a set of equations that describe the process. Next, there is a criterion, or objective, or an index of performance of the control system to be satisfied. Associated with the equations of the process is a control or forcing function that is manipulated to satisfy the criterion within the constraints imposed on the system. In a general sense, all dynamic programming problems satisfy this theme and they may very properly be called control problems.

Without a doubt, the most widely used tool of the servo engineer is the Laplace transform. The utility of the Laplace transform is based on the fact that ordinary linear differential equations with constant coefficients can be solved explicitly in terms of exponential functions. The additivity of the solutions of ordinary linear differential equations, the steady state and the transient solution, are all neatly handled with the Laplace transform.

For linear systems, the control engineer uses various analytical devices based on the Laplace transform. These include the Bode plot, the Nyquist plot, and the root-locus plot. These plots, used either individually, or in concert with the others, provide tools for the design and analysis of linear systems.
For nonlinear systems, the Laplace transform is less powerful than for linear systems, since the principle of superposition does not apply. The control engineer has no one single powerful tool for coping with nonlinear systems. He may use the describing function method or the phase-plane representation of the system characteristics.

Dynamic programming presents another tool for the control engineer. It offers a uniform way to treat problems whether linear or nonlinear and, in addition, can be used to solve problems that have previously defied solution. In particular, control problems involving inequality constraints and/or the absolute value of functions, which often immobilize conventional techniques, are handled very naturally by dynamic programming.

In the next sections, we will discuss, from the dynamic programming point of view, deterministic control problems of the following types:

(a) averaging control problems
(b) terminal control problems
(c) minimum of maximum control problems
(d) time lag control problems

In Chapter 9, Stochastic Processes, we will present some stochastic and adaptive control problems.

We will first discuss problems (a)–(d) with respect to general control problems and then discuss typical chemical engineering problems in the next chapter.

To review the chapter briefly, in Section 2 we note the relationship between the feedback concept and the multi-stage decision process. In Sections 3–6, averaging control problems are presented using various criteria subject to Van der Pol's equation. An equally important class of control problems, namely, terminal control problems, are described in Sections 7–9. A generalization of the terminal control problem is shown in Section 10 to embrace the averaging control problem. A criterion unfamiliar to chemical engineers, the minimization of a maximum deviation, is employed in Section 11. The resolution of a terminal control problem with time lags is the subject of Section 12. An alternate approach to this problem via the solution of a linear integral equation is described next. The relative merits of each approach are pointed out. Finally, in Sections 14 and 15, the utilization of dynamic programming in the optimal design of a control system is shown together with a numerical example.
2. Feedback Control and Multi-Stage Decision Process

To set the stage for what follows, we claim that feedback control may be looked upon as a multi-stage decision process and, conversely, multi-stage process may be looked upon as feedback processes.

In typical feedback control, an output signal is compared to a reference signal and the sign and the magnitude of the error signal between them is used to make compensating control action. The information pattern here is the receiving and comparing of two instantaneous signals, the result of which is an instantaneous decision to take a control action triggered by the instantaneous error signal.

In a multi-stage decision process, the state variables and the number of stages remaining correspond to the error signal in feedback control. The decision at any stage depends only on the state of the system at that stage and the number of stages remaining. The past history of the process has no influence whatsoever on the decision being made. In other words, the future course of the process depends only on the current state and the number of remaining stages.

The feedback principle is implicit in the Principle of Optimality. The Principle of Optimality requires that whatever the initial state (the initial error signal), a sequence of control actions will be taken over the remainder of the process to guide it optimally over the remainder of the process.

3. Averaging Control, General Formulation

A number of control problems may be cast into the form of maximizing or minimizing a function over time. These problems can be designated as averaging control problems. Typical problems may be cast into the form of minimizing \[ \int_0^T (x^d - x) \, dt \] where \( x^d \) is the desired state and \( x \) is the actual state. Another form that appears quite often is minimizing \[ \int_0^T (x^d - x)^2 \, dt. \]

To generalize, let us consider the problem of minimizing \( J(y) \) as defined by

\[
J(y) = \int_0^T F(x, y) \, dt
\]  

where \( x(t) \) and \( y(t) \) are connected by the relationship

\[
\frac{dx}{dt} = G(x, y)
\]
and the initial condition is

\[ x(0) = c \]  

(3)

The functions \((x^d - x)\) and \((x^d - x)^2\) correspond to the function \(F(x, y)\). The minimization is accomplished by finding the proper \(y(t)\); that is to say, \(y(t)\), the control vector, must be manipulated to minimize \(J(y)\).

This problem is identical to those discussed in Chapter 4, Dynamic Programming and the Calculus of Variations, and is handled by the same techniques.

As before, it is convenient to cast the problem into a discrete form by dividing the interval \((0, T)\) into \(N\) equal subintervals of duration \(\Delta\), so that \(N\Delta = T\). We count time backward so that stage \(N\) corresponds to the beginning of the process and stage 0 corresponds to the end. The time scale is given in Fig. 1.

\[
\begin{array}{c}
\text{Beginning} \\
0 \\
N \rightarrow \text{Stages} \\
\text{End} \\
T
\end{array}
\]

FIG. 1.

The functions \(y(t)\) and \(x(t)\) then become \(y(k\Delta)\) and \(x(k\Delta)\). For convenience we write \(y_k = y(k\Delta)\) and \(x_k = x(k\Delta)\).

Equations (1) and (2) then are cast into the following form:

\[
J_N(y_k) = \sum_{k=1}^{N} F(x_k, y_k) \Delta
\]

(4)

\[
\frac{x_{k+1} - x_k}{\Delta} = G(x_k, y_k), \quad k = 0, 1, 2, \ldots, N
\]

(5)

By virtue of the time scale convention adopted, the initial condition, Eq. (3), is given by

\[ x_N = c \]

(6)

Let us define

\[
f_N(c) = \text{Min} J_N(y_k) = \text{Min} \sum_{k=1}^{N} F(x_k, y_k) \Delta = \text{the minimum value of the summation, } J_N(y_k) \text{ over the } N \text{ remaining stages of the process starting with system initially in state } c, \text{ and using an optimal policy.}
\]
4. AVERAGING CONTROL, EXAMPLE 1

The solution to the minimization problem is described by

$$f_N(c) = \min_{y_N} [F(c, y_N) \Delta + f_{N-1}(c + G(c, y_N) \Delta)]$$ (7)

$$f_1(c) = \min_{y_1} [F(c, y_1) \Delta]$$ (8)

We note in Eq. (8) for the one-stage process that given the initial value of \( x = x_1 = c \), the choice of \( y_1 \) must be made to minimize \( F \).

In the \( N \)-stage process, the proper choice of \( y_N \) during stage \( N \) minimizes the right-hand side of Eq. (7). Having found the proper choice of \( y_N \), we observe that the new initial state for the remaining \((N - 1)\) stages is \( c + G(c, y_N) \Delta \). This, of course, is merely the solution of Eq. (5) for \( x_{N-1} \). Given the number of stages remaining and the initial state \( c \), the functional equations describe the minimization process. In this process the choice of \( Y_k \) determines how \( x_k \) changes with time and both \( x_k \) and \( Y_k \) determine the values of the functions \( G \) and \( f_N(c) \).

We should point out here that this problem in this simple form could be solved by the conventional calculus of variations approach. The inclusion of certain constraints on the \( Y_k \), such as \( b_2 < Y_k \leq b_1 \), where \( b_1 \) and \( b_2 \) are constants, would, however, in general create grave difficulties to the calculus of variations approach.

The casting of this control problem into the dynamic programming form is particularly useful in evaluating numerical solutions. It would have been possible as described in Chapters 4 and 5 to have developed a partial differential equation from the original formulation. The above method is much more direct and in general seems like a simpler approach.

4. Averaging Control, Example 1

We will now consider some specific examples of the averaging control problem. We consider the problem of minimizing \( J(w) \):

$$J(w) = \int_0^T (x^2 + x'^2 + aw^2) \, dt$$ (1)

where \( x \) satisfies Van der Pol's equation

$$x'' + \mu(x^2 - 1)x' + x = w(t)$$ (2)

with the forcing function \( w(t) \), and where \( \mu \) and \( a \) are known constants.
The initial conditions are

\[ x(0) = c_1, \quad x'(0) = c_2 \]  \hspace{1cm} (3)

The term \( w(t) \) is the forcing function that is manipulated to achieve the desired control. The coefficient \( a \) may be considered to be a measure of the cost of using the forcing function \( w(t) \).

As before, we cast the problem into a discrete form, where the time span \((0, T)\) is divided into \( N \) equal subintervals of length \( \Delta \), so \( N \Delta = T \). We count time backward as in the previous example. We write \( x_k \) for \( x(k \Delta) \), \( y_k \) for \( y(k \Delta) \), and \( w_k \) for \( w(k \Delta) \).

It is convenient to let

\[ y_k = x_k' = \frac{x_{k+1} - x_k}{\Delta}, \quad k = 0, 1, 2, \ldots, N \]  \hspace{1cm} (4)

or

\[ x_{k+1} = x_k + y_k \Delta \]  \hspace{1cm} (5)

The discrete form of the equations are

\[ J_N(w_k) = \sum_{k=1}^{N} (x_k^2 + y_k^2 + aw_k^2) \Delta \]  \hspace{1cm} (6)

\[ \frac{y_{k+1} - y_k}{\Delta} + \mu(x_k^2 - 1) y_k + x_k = w_k \]  \hspace{1cm} (7)

\[ y_{k+1} = y_k + [w_k - \mu(x_k^2 - 1) y_k - x_k] \Delta \]  \hspace{1cm} (8)

The initial conditions by our time convention are given by

\[ x_N = c_1, \quad y_N = c_2 \]  \hspace{1cm} (9)

Let us define

\[ f_N(c_1, c_2) = \min_{w_k} J_N(w_k) = \text{the minimum of } J_N(w_k) \text{ over the } N \text{ remaining time stages of the process, starting in the known state } (c_1, c_2), \text{ subject to Eqs. (5), (8), and (9) and using an optimal policy.} \]  \hspace{1cm} (10)

The functional equations describing the control process are

\[ f_N(c_1, c_2) = \min_{w_N} \left[ (c_1^2 + c_2^2 + aw_N^2) \Delta + f_{N-1}(c_1 + c_2 \Delta, c_2 \right. \right. \]

\[ + \left. \left. \left. \{w_N - \mu(c_1^2 - 1) c_2 - c_1 \Delta] \right) \right) \right) \]  \hspace{1cm} (11)

\[ f_1(c_1, c_2) = \min_{w_1} \left[ c_1^2 + c_2^2 + aw_1^2 \right] \]  \hspace{1cm} (12)

\[ f_1(c_1, c_2) = (c_1^2 + c_2^2) \Delta \]  \hspace{1cm} (13)
In this formulation, we observe that to minimize $f_1(c_1, c_2)$, we set $w_1 = 0$. The new $c_1$ and $c_2$ in Eq. (11) for $f_N(c_1, c_2)$ are found by Eqs. (5) and (8) where the "old" $c_1$ and $c_2$ correspond to $x_k$ and $y_k$ and the "new" $c_1$ and $c_2$ correspond to $x_{k+1}$ and $y_{k+1}$.

In this problem, the values that $w_k$ assumes are constrained by the factor $a$, which represents the cost for the privilege of using $w_k$. In some problems, a hierarchy of costs might be set up so that the cost of $w_k$ is a function of the value of $w_k$ itself. For example, the cost structure might appear as

$$
\begin{align*}
    a_1, & \quad 0 \leq w_k < 1 \\
    a_2, & \quad 1 \leq w_k < 2 \\
    a_3, & \quad 2 \leq w_k < 3
\end{align*}
$$

where $a_1 > a_2 > a_3$.

In this case, the problem formulation is identical to that given above. The numerical evaluation, however, is more involved due to the cost structure.

5. Averaging Control, Example 2

In this section, we discuss a variation of the problem in Section 4. The principle difference is that the control vector $w(t)$ is restricted by a limit on its absolute value $|w(t)|$. We desire to minimize $J(w)$:

$$
J(w) = \int_0^T (x^2 + x'^2) \, dt
$$

subject to

$$
\begin{align*}
    x'' + \mu(x'^2 - 1) x' + x &= w(t) \\
    x(0) &= c_1; \quad x'(0) = c_2 \\
    |w(t)| &\leq 1
\end{align*}
$$

The equations are cast into discrete form as above:

$$
J_N(w_k) = \sum_{k=1}^{N} (x_k^2 + y_k^2) \Delta
$$

$$
\begin{align*}
    x_{k+1} &= x_k + y_k \Delta \\
    y_{k+1} &= y_k + \Delta[w_k - \mu(x_k^2 - 1) y_k - x_k]
\end{align*}
$$

The initial conditions are given by

$$
\begin{align*}
    x_N &= c_1; \quad y_N = c_2 \\
    |w_k| &\leq 1
\end{align*}
$$
We define
\[ f_N(c_1, c_2) = \text{Min}_{|w_k| \leq 1} \sum_{k=1}^{N} (x_k^2 + y_k^2) \Delta = \text{the minimum of } J_N(w_k) \text{ over the } N \text{ remaining stages of the process starting in state } (c_1, c_2), \text{ subject to (6)-(9), and using an optimal policy} \]

The functional equations are
\[ f_N(c_1, c_2) = \text{Min}_{|w_k| < 1} [(c_1^2 + c_2^2) \Delta + f_{N-1}(c_1 + c_2 \Delta, c_2 + \Delta(w_N - \mu(c_1^2 - 1,c_2 - c_1)))] \]
\[ f_1(c_1, c_2) = \text{Min}_{|w_1| < 1} [c_1^2 + c_2^2] \Delta = (c_1^2 + c_2^2) \Delta \]

In this formulation, \( |w_k| \) is constrained to be less than 1. While this type of a constraint would have been extremely difficult to handle using the formal calculus of variations, it offers no problem in dynamic programming. In fact, the constraint serves to narrow the possible range of values which must be searched to find the optimal \( w_k \).

We observe that the formulation and equations of this section are quite similar to the previous section. In fact, the only ostensible difference between Eq. (11) of this section and Eq. (11) of the previous section is the term \( aw_k^2 \). However, in the numerical evaluation of Eq. (10) of this section, the constraint, Eq. (9), is honored. Compare the definitions of \( f_N(c_1, c_2) \) in Sections 4 and 5.

Another possible way of solving this problem is to use the cost \( a \), when \( |w(t)| \leq 1 \) and \( b \) when \( |w(t)| > 1 \) where \( b > a \).

We then define
\[ j(w) = \begin{cases} \int_{0}^{T} [x^2 + x'^2 + a |w|] dt, & |w| \leq 1 \\ \int_{0}^{T} [x^2 + x'^2 + b |w|] dt, & |w| > 1 \end{cases} \]

Following the same lines, we write
\[ f_N(c_1, c_2) = \text{Min}_{|w_k|} \int_{N}(w_k) \]
\[ f_N(c_1, c_2) = \text{Min}_{|w_N|} \begin{cases} [(c_1^2 + c_2^2) + a |w| \Delta + f_{N-1}(c_1 + c_2 \Delta, c_2) + \Delta(w_N - \mu(c_1^2 - 1,c_2 - c_1))] \leq 1 \\ [(c_1^2 + c_2^2) + b |w| \Delta + f_{N-1}(c_1 + c_2 \Delta, c_2) + \Delta(w_N - \mu(c_1 - 1,c_2 - c_1))] > 1 \end{cases} \]
\[ f_1(c_1, c_2) = (c_1^2 + c_2^2) \Delta \]
In this formulation, the cost structure forces \( w_k \) to lie within the bound \( |w_k| \leq 1 \).

6. Averaging Control, Example 3

Let us consider a variation of the problem in Section 4. We desire to minimize

\[
J(w) = \int_0^T [(x - x^d)^2 + (x' - x'^d)^2 + aw^2] \, dt
\]  

subject to

\[
x'' + \mu(t)(x^d - 1)x' + x = w(t)
\]  

\[
\int_0^T w(t) \, dt \leq B
\]  

(3)

\[
\mu_{\text{Min}} \leq \mu(t) \leq \mu_{\text{Max}}
\]  

(4)

where \( \mu(t) \) is a parameter to be manipulated.

\[
x(0) = c_1; \quad x'(0) = c_2
\]  

(5)

The terms \( a, x^d, x'^d, \mu_{\text{Min}}, \) and \( \mu_{\text{Max}} \), are known constants.

In this problem, we wish to minimize the error squared of the variable \( x \) about a known desired value of \( x \), namely, \( x^d \), plus the error squared of the derivative around a known desired value of \( x' \), namely, \( x'^d \). In addition to the cost of using \( w(t) \), we have an isoperimetric constraint, Eq. (3), on the total quantity of \( w(t) \) that may be expended. The \( w(t) \) is constrained, therefore, by the actual cost of using it, as well as by a limited resource.

We minimize over two variables by permitting both \( w(t) \) to be adjusted and the parameter \( \mu(t) \) in Eq. (2) to be manipulated over the range given by Eq. (4).

To take care of the isoperimetric constraint, we introduce the Lagrangian multiplier \( \lambda \) and form

\[
J(w) = \int_0^T [(x - x^d)^2 + (x' - x'^d)^2 + aw^2 + \lambda w] \, dt
\]  

(6)

As before, we define

\[
f_N(c_1, c_2) = \min_{w_k, \mu_k} J_N(w_k)
\]  

(7)
The functional equations are

\[
\begin{align*}
 f_N(c_1, c_2) &= \min_{w_N, \mu_N} \left[ ((c_1 - x_d)^2 + (c_2 - x_d')^2 + ax^2 + \lambda w_N) \Delta \right. \\
 &\quad + f_{N-1}(c_1 + c_2 \Delta, c_2 + \{w_N - \mu_N(c_1^2 - 1) c_2 - c_1 \} \Delta) \\
 f_1(c_1, c_2) &= [(c_1 - x_d)^2 + (c_2 - x_d')^2] \Delta
\end{align*}
\]  

(8) (9)

The optimization over two variables \( \mu_k \) and \( w_k \) requires more searching for the minimum than one variable. The basic approach given in Section 3 still holds. The determination of the proper value for the Lagrangian multiplier \( \lambda \) is a trial and error procedure, which has been described in Chapter 5, Section 19.

7. Terminal Control Problems

In many processes, it is desired to maximize or minimize a function at the final time. How the process behaves over the span from the initial time to the final time is of no interest in itself. The largest (or smallest) value of the function at terminal conditions is desired regardless of the path taken to achieve this goal.

This is in direct contrast to the averaging control problems discussed in the previous sections where the variables-time path, as well as the objective, is important over the span \((0, T)\).

Let us consider a function \( H(x(t)) \) whose final value, \( H(x(T)) \), we desire to maximize. We wish to maximize \( J(y) \):

\[
J(y) = H(x(T))
\]  

(1)

subject to

\[
\frac{dx}{dt} = G(x, y)
\]  

(2)

where \( x \) and \( y \) are scalars. The function \( J(y) \) is constrained by

\[
\int_0^T L(x, y) \, dt = L_1
\]  

(3)

where \( L_1 \) is a constant. The initial condition is

\[
x(0) = c
\]  

(4)

and the maximization is achieved by manipulating the variable \( y(t) \).
We redefine $J(y)$ as

$$J(y) = H(x(T)) + \lambda \int_0^T L(x, y) \, dt$$

(5)

where $\lambda$ is a Lagrangian multiplier.

As before, we break the interval $(0, T)$ into $N$ equal increments of $\Delta$ duration so $N\Delta = T$. We count time backward so that $N = N$ corresponds to the initial time and $N = 0$ corresponds to the final time, as given in Fig. 1.

Equations (2) and (3) may be put into finite difference form:

$$x_{k+1} - x_k = G(x_k, y_k) \Delta, \quad k = 0, 1, 2, ..., N$$

(6)

$$\sum_{k=1}^{N} L(x_k, y_k) \Delta = L_1$$

(7)

In accordance with our time convention, the initial condition is

$$x_N = c$$

(8)

We define

$$f_N(c) = \text{the maximum value of } J_N(y_k) \text{ over the } N \text{ remaining states of the process, starting in state } c, \text{ subject to Eqs. (6)-(8), and using an optimal policy}$$

$$f_N(c) = \text{Max } J_N(y_k) = \text{Max } y_k \left[ H(x_0) + \lambda \sum_{k=1}^{N} L(x_k, y_k) \Delta \right]$$

(9)

where $x_0$ is the final value of $x$ corresponding to $x(T)$;

$$f_N(c) = \text{Max } \left[ \lambda L(c, y_N) \Delta + f_{N-1}(c + G(c, y_N) \Delta) \right]$$

(10)

$$f_1(c) = \text{Max } \left[ \lambda L(c, y_1) \Delta + H(c + G(c, y_1) \Delta) \right]$$

(11)

where

$$x_0 = c + G(c, y_1)\Delta$$

(12)

Another way to write Eq. (11) is

$$f_1(c) = \text{Max } \left[ \lambda L(c, y_1) \Delta + f_0(c + G(c, y_1) \Delta) \right]$$

(13)

where

$$f_0(c + G(c_1, y_1) \Delta) = H(c + G(c_1, y_1) \Delta) = H(x_0)$$

(14)
From the definition of $f_N(c)$, we see that the function $f_0(c)$ is the maximum value of $J_N(y_k)$ over the zero remaining stages of the process. A process with zero remaining stages is, of course, at its final or terminal point.

8. Terminal Control Problems in Two Variables

Let us consider a terminal control problem in two variables, $x_1(t)$ and $x_2(t)$. The objective is to maximize

$$J(y_1, y_2) = \max H[x_1(T), x_2(T)]$$

subject to

$$\frac{dx_1}{dt} = G_1(x_1, x_2, y_1, y_2)$$

$$\frac{dx_2}{dt} = G_2(x_1, x_2, y_1, y_2)$$

$$\int_0^T L(x_1, x_2, y_1, y_2) \, dt = L_1$$

$$x_1(0) = c_1, \quad x_2(0) = c_2$$

We define

$$f_N(c_1, c_2) = \max J_N[(y_1)_k, (y_2)_k]$$

Following the procedure in Section 7, and counting time backward, we have

$$f_N(c_1, c_2) = \max \left\{ \frac{\lambda L(c_1, c_2, (y_1)_N, (y_2)_N) \Delta}{(y_1)_N, (y_2)_N} \right\}$$

We write $(y_1)_k$ for $y_1(k\Delta)$, $(y_2)_k$ for $y_2(k\Delta)$, $(x_1)_k$ for $x_1(k\Delta)$, and $(x_2)_k$ for $x_2(k\Delta)$.

$$f_N(c_1, c_2) = \max \left\{ \frac{\lambda L(c_1, c_2, (y_1)_N, (y_2)_N) \Delta}{(y_1)_N, (y_2)_N} \right\}$$

$$+ f_{N-1}(c_1 + G_1(c_1, c_2, (y_1)_N, (y_2)_N) \Delta, c_2 + G_2(c_1, c_2, (y_1)_N, (y_2)_N) \Delta)$$

$$f_1(c_1, c_2) = \max \left\{ \frac{\lambda L(c_1, c_2, (y_1)_1, (y_2)_1) \Delta}{(y_1)_1, (y_2)_1} \right\}$$

$$+ f_0(c_1 + G_1(c_1, c_2, (y_1)_1, (y_2)_1) \Delta, c_2 + G_2(c_1, c_2, (y_1)_1, (y_2)_1) \Delta)$$

where

$$f_0(c_1 + G_1(c_1, c_2, (y_1)_1, (y_2)_1) \Delta, c_2 + G_2(c_1, c_2, (y_1)_1, (y_2)_1) \Delta) = H((x_1)_0, (x_2)_0)$$
9. Multi-Variable Terminal Control

In some multi-variable problems it is required to maximize a terminal performance index which contains only some of the multiple variables. For example, we may be required to maximize the function $J(y_1, y_2)$, where

$$J(y_1, y_2) = H[(x_1(T), x_2(T))]$$

subject to the equations

$$\frac{dx_1}{dt} = G_1(x_1, x_2, x_3, y_1, y_2, y_3, y_4)$$

$$\frac{dx_2}{dt} = G_2(x_1, x_2, x_3, y_1, y_2, y_3, y_4)$$

$$\frac{dx_3}{dt} = G_3(x_1, x_2, x_3, y_1, y_2, y_3, y_4)$$

$$\frac{dx_4}{dt} = G_4(x_1, x_2, x_3, y_1, y_2, y_3, y_4)$$

$$\int_0^T L(x_1, x_2, x_3, y_1, y_2, y_3, y_4) = L_1$$

with the initial conditions

$$x_1(0) = c_1, \quad x_2(0) = c_2, \quad x_3(0) = c_3, \quad x_4(0) = c_4$$

In this case, we have the option of setting up the problem as before in Sections 7 and 8, and defining an objective function in four state variables, $f_N(c_1, c_2, c_3, c_4)$. We also have the option of setting it up by the procedure given in Chapter 5, Sections 27–30 by defining a function $f_N(b_1, b_2)$. The $b_1$ and $b_2$ are the solutions to the homogeneous linear equations for $x_1$ and $x_2$ that approximate the $G_i$ in Eqs. (2) and (3). We, of course, always choose to reduce dimensionality where possible.

10. Generalization of Terminal Control

In our discussions of averaging control and terminal control problems, we treated them as separate entities. In the one case, the optimization was carried out over the entire time interval. In the other case, the optimization was concerned only with satisfying terminal conditions.

By means of a artifice, we can convert an averaging control problem into a terminal control problem.
Suppose that we desire to minimize
\[ J(w) = \int_0^T F(x_1, x_2, ..., x_N, w_1, w_2, ..., w_M) \, dt \]  
(1)

where \( x_1, x_2, ..., x_N \) are components of a vector \( x \) and \( w_1, w_2, ..., w_M \) are components of the forcing function vector \( w \). We introduce a new variable
\[ x'_{N+1}(t) = F(x_1, x_2, ..., x_N, w_1, w_2, ..., w_M) \]  
(2)

and the boundary condition
\[ x_{N+1}(0) = 0 \]  
(3)

The functional \( J(w) \) now may be expressed as
\[ J(w) = \int_0^T x'_{N+1}(t) \, dt = x_{N+1}(T) - x_{N+1}(0) \]  
(4)

Using the boundary condition Eq. (3), we have
\[ J(w) = x_{N+1}(T) \]  
(5)

The minimization of the original averaging control problem Eq. (1) has been transformed into the terminal control problem of minimizing \( x_{N+1}(T) \).

11. Minimum of Maximum Deviation

At times it is very important to control a process so that the maximum deviation of a quantity from a prescribed level is minimized over a time interval. This problem which cannot be attacked very well by the calculus of variations can be handled by dynamic programming.

Let us first make the following critical observation. Suppose we have a sequence of numbers \((A_1, A_2, ..., A_n)\) and suppose we desire to find the largest number, namely, \( A_1 \). The largest number may be found by partitioning the set of \( A \)'s as follows:
\[ \text{Max} (A_1, A_2, ..., A_n) = \text{Max} [A_1, \text{Max} (A_2, A_3, ..., A_n)] \]  
(1)

This result may be readily established inductively.

Consider the problem of minimizing
\[ J(w) = \max_{0 \leq t \leq T} |x(t)| \]  
(2)
subject to

\[ x'' + \mu(x^2 - 1) x' + x = w(t) \]  
\[ x(0) = c_1 \]  
\[ x'(0) = c_2 \]  
\[ |w(t)| \leq 1 \]  

We desire to give the following interpretation to \(|x(t)|\). If we consider as a datum \(x(0) = 0\), then

\[ |x(t) - 0| = |x(t)| = \text{the absolute value of the deviation from the datum} \]  

We can formulate this problem in discrete form as follows:

\[ J_N(w_k) = \max_{1 \leq k \leq N} |x_k| \]  
\[ x_{k+1} = x_k + y_k \Delta, \quad k = 0, 1, 2, ..., N \]  
\[ y_{k+1} = y_k + \Delta[w_k - \mu_k(x_k^2 - 1)y_k - x_k] \]  
\[ x_N = c_1; \quad y_N = c_2 \]  
\[ f_N(c_1, c_2) = \min_{w_k} J_N(w_k) = \min_{w_k} \max_{1 \leq k \leq N} |x_k| \]  

In words

\[ f_N(c_1, c_2) = \text{the minimum of the maximum of the absolute value of the deviation} \]  
\[ x_k \text{ over the } N \text{ remaining stages of the process beginning in state} \]  
\[ (c_1, c_2), \text{ and using an optimal policy} \]  

\[ f_N(c_1, c_2) = \min_{w_k} \max_{1 \leq k \leq N} |x_k| \]  

Using the partitioning principle exhibited in Eq. (1), we write

\[ f_N(c_1, c_2) = \min_{w_k} \max \left[ \left| x_N \right|, \max_{1 \leq k \leq N-1} |x_k| \right] \]  
\[ f_N(c_1, c_2) = \min_{w_k} \max \left[ \left| x_N \right|, J_{N-1}(w_k) \right] \]  
\[ f_N(c_1, c_2) = \max \left[ \left| x_N \right|, \min_{w_k} J_{N-1}(w_k) \right] \]  
\[ f_N(c_1, c_2) = \max \left[ |c_1|, f_{N-1}(c_1 + c_2 \Delta, c_2 + \{w_N - \mu_N(c_1^2 - 1)c_2 - c_1\} \Delta) \right] \]  
\[ f_1(c_1, c_2) = |c_1| \]  

In Eq. (15), the regrouping is carried out as explained in Eq. (1). Equation (16) follows from the definition of \(J_N(w_k)\) in (8). In Eq. (17), the Min operation is brought inside the brackets because the minimization is taken over \(w_k\) over the interval \(1 \leq k \leq N - 1\). As a consequence \(w_k\) has no affect on \(x_N\), but affects only \(J_{N-1}(w_k)\). The choice of \(w_k\) determines the new value of \(c_2\).
12. Control Problem with Time Lags

In many problems, there is a time lag between the manipulation of the control vector and the response of the system to this control action. The time lag in a sense is a measure of the inertia of the system to respond. In systems with time lags, especially long time lags, it is very important to make correct decisions since the results of improper control decisions in these systems are particularly difficult to correct, due to the slowness of the system response.

We may handle time lag problems by dynamic programming, but we pay a price for the time lag by dealing with a multi-dimensional problem.

Let us consider a time-lag problem described by the linear differential difference equation

$$\frac{dx(t)}{dt} = ax(t - T_0) + v(t)$$

(1)

The equation states that the rate of change in \( x(t) \) is directly proportional to the value of \( x \), at \( T_0 \) time units earlier. The term \( v(t) \) is the control variable.

The \( v(t) \) term is to be chosen so that

$$\int_0^T v^2(t) \, dt \leq V$$

(2)

where \( V \) is a constant.

For terminal control problems, \( v(t) \) must be chosen so \( x^2(t) \) reaches a desired terminal value \( x^2(T) \).

The control objective is to minimize \( J(v) \):

$$J(v) = \lambda \int_0^T v^2 \, dt + x^2(T)$$

(3)

where \( \lambda \) is a Lagrangian multiplier.

As before, in non-time-lag problems, the control period \((0, T)\) is divided into \( N \) equal increments of \( \Delta \) duration so \( N\Delta = T \). In addition, here we must consider the history of the process with \( M\Delta = (t - T_0) \) units of delay, as shown in the time scale in Fig. 2.

The equations may now be written in finite difference form as

$$\frac{x_{k-1} - x_k}{\Delta} = ax_{k+M} + v_k, \quad k = 0, 1, 2, ..., N$$

(4)
where \( x_{k+M} = x((k+M)\Delta) \) and corresponds to \( x \) at time \((t - T_0)\) in Eq. (1).

\[
x_{k-1} = x_k + \Delta(ax_{k+M} + v_k) \tag{5}
\]

The function to be minimized is \( J_N(v_k) \):

\[
J_N(v_k) = \lambda \sum_{k=1}^{N} v_k^2 + x_0^2 \tag{6}
\]

The initial conditions are given by

\[
x_N = c_0, \quad x_{N+1} = c_1, \ldots, x_{N+M} = c_M \tag{7}
\]

where the current value of \( x_N = c_0 \) and the historical values of the \( x \) terms are given by \( c_1, c_2, \ldots, c_M \).

In this process, the state of the system is specified by the current \( x \), that is \( x_N \), and all the historical \( x \)'s back \( M\Delta \) time units of lag. In particular, the state is specified by

\[
S = S(x_{N+M}, x_{N+M-1}, x_{N+M-2}, \ldots, x_N) \tag{8}
\]

or using (7)

\[
S = S(c_M, c_{M-1}, c_{M-2}, \ldots, c_0) \tag{9}
\]

The need for specifying the historical \( x \)'s, as well as the current \( x \), namely \( x_N \), is seen from Eq. (1) or Eq. (5). In these equations, there is a relation between \( x_{N-1}, x_N, \) and \( x_{N+M} \). For the next time step, the equation relates \( x_{N-2}, x_{N-1}, x_{N+M-1} \). If we continue this way, it readily becomes apparent that, to pass from one stage to the succeeding one, the historical record of the \( x \)'s is required. Since by definition a state contains all the information required to make the transition, the quantities \( (c_M, c_{M-1}, c_{M-2}, \ldots, c_0) \) must be specified.
We now define

\[ f_N(S) = f_N(c_M, c_{M-1}, c_{M-2}, \ldots, c_0) = \text{the minimum return over the } N \text{ remaining stages for the system beginning in state } c_M, c_{M-1}, \ldots, c_0, \text{ and following an optimal policy} \]

\[ f_N(S) = \min_{v_k} \left[ \lambda \sum_{k=1}^{N} v_k^2 + x_0^2 \right] \]  

(11)

\[ f_N(S) = \min_{v_k} \left[ \lambda v_k^2 + \lambda \sum_{k=1}^{N-1} v_k^2 + x_0^2 \right] \]

(12)

\[ f_N(c_M, c_{M-1}, \ldots, c_0) = \min_{v_N} \left[ \lambda v_N^2 + f_{N-1}(c_{M-1}, c_{M-2}, \ldots, c_0, c_0 + \Delta(ac_M + v_N)) \right] \]  

(13)

\[ f_1(c_M, c_{M-1}, \ldots, c_0) = \min_{v_1} \left[ \lambda v_1^2 + (c_0 + \Delta(ac_M + v_1))^2 \right] \]

(14)

\[ f_1(c_M, c_{M-1}, \ldots, c_0) = \min_{v_1} \left[ \lambda v_1^2 + x_0^2 \right] \]

(15)

where

\[ x_0 = c_0 + \Delta(ac_M + v_1) \]  

(16)

In Eq. (13), we note that the decision to choose the value of \( v_N \) that minimizes, transforms the current value of \( x_N = c_0 \) for stage \( N \) into \( c_0 + \Delta(ac_M + v_N) \) by virtue of Eq. (5). The other terms in the state description are not affected by the choice of \( v_N \), but are shifted forward in time by one \( \Delta \) unit. As a consequence, the last state variable \( c_M \) is dropped. The state of the system for the \((N-1)\)-stage process is described by

\[ S = S(c_{M-1}, c_{M-2}, \ldots, c_0, c_0 + \Delta(ac_M + v_N)) \]

(17)

By virtue of the imbedding process and by virtue of Eq. (7) where \( N \) is replaced by \((N-1)\), the state of the system is described by the initial values \( c_M, c_{M-1}, \ldots, c_0 \), which are redefined for the \((N-1)\)-stage process. Therefore, we see that the new state variables in stage \((N-1)\) correspond to the “old” state variables in stage \( N \) as follows:

<table>
<thead>
<tr>
<th>New</th>
<th>Old</th>
</tr>
</thead>
<tbody>
<tr>
<td>( c_M )</td>
<td>( c_{M-1} )</td>
</tr>
<tr>
<td>( c_{M-1} )</td>
<td>( c_{M-2} )</td>
</tr>
<tr>
<td>( \vdots )</td>
<td>( \vdots )</td>
</tr>
<tr>
<td>( c_1 )</td>
<td>( c_0 )</td>
</tr>
<tr>
<td>( c_0 )</td>
<td>( c_0 + \Delta(ac_M + v_N) )</td>
</tr>
</tbody>
</table>

(18)
In this time lag problem, the state of the system is specified by \((M+1)\) state variables. In contrast to the previous multi-dimensional state variable problems, for the time lag problem, only one of the state variables, namely \(c_0\), is transformed by the choice of the \(v_N\) by the differential equation of the process. The other state variables are shifted by one \(\Delta\) unit no matter what value \(v_N\) takes. These are the principle points of novelty in the time lag problem.

One way to solve Eqs. (13) and (14) is to use differential calculus. For example, differentiating Eq. (14) with respect to \(v_1\) and setting the derivative equal to zero yields the value of minimizing value of \(v_1\). Substitution of the minimizing value of \(v_1\) into Eq. (14) yields \(f_1(c_M, c_{M-1}, \ldots, c_0)\). In a similar manner, \(v_2\) and \(f_2\) are found, and sequentially all the \(v_k\) and \(f_k\) terms.

To be specific, we will develop the \(v_1, f_1, v_2,\) and \(f_2\) in detail and then give the general form for \(v_k\) and \(f_k\).

We start at the last stage where

\[
f_1(c_M, c_{M-1}, \ldots, c_0) = \min_{v_1} \left[ \lambda v_1^2 + \left( c_0 + \Delta(ac_M + v_1) \right)^2 \right] \tag{19}
\]

Taking the derivative of the right-hand side with respect to \(v_1\) and setting it equal to zero, we find

\[
v_1 = \frac{-\Delta(c_0 + \Delta ac_M)}{\lambda + \Delta^2} \tag{20}
\]

Using this value of \(v_1\), we have

\[
f_1(c_M, c_{M-1}, \ldots, c_0) = \frac{\lambda}{\lambda + \Delta^2} [c_0 + \Delta ac_M]^2 \tag{21}
\]

Now for the two-stage process

\[
f_2(c_M, c_{M-1}, \ldots, c_0) = \min_{v_2} \left[ \lambda v_2^2 + f_1(c_{M-1}, c_{M-2}, \ldots, c_0 + \Delta(ac_M + v_2)) \right] \tag{22}
\]

By Eq. (21) we write

\[
f_1(c_{M-1}, c_{M-2}, \ldots, c_0 + \Delta(ac_M + v_2)) = \frac{\lambda}{\lambda + \Delta^2} [c_0 + \Delta(ac_M + v_2) + \Delta c_{M-1}]^2 \tag{23}
\]

where the \(c_0\) of Eq. (21) corresponds to the \(c_0 + \Delta(ac_M + v_2)\) of Eq. (23) and the \(c_M\) of Eq. (21) corresponds to the \(c_{M-1}\) of Eq. (23).
Substituting Eq. (23) into Eq. (22), we have

$$f_2(c_M, c_{M-1}, ..., c_0) = \min_{t_2} \left[ \frac{\lambda}{\lambda + \Delta^2} [c_0 + \Delta(ac_M + v_2) + \Delta ac_{M-1}]^2 \right]$$  \hspace{1cm} (24)

By differentiating with respect to $v_2$ the right-hand side of Eq. (24) and setting the derivative equal to zero, we find

$$v_2 = -\frac{\lambda}{\lambda + 2\Delta^2} [c_0 + \Delta ac_M + \Delta ac_{M-1}]$$  \hspace{1cm} (25)

$$f_2(c_M, c_{M-1}, ..., c_0) = \frac{\lambda}{\lambda + 2\Delta^2} [c_0 + \Delta ac_M + \Delta ac_{M-1}]^2$$  \hspace{1cm} (26)

Continuing into this manner, we can show that

$$v_3 = -\frac{\Delta}{\lambda + 3\Delta^2} [c_0 + \Delta a(c_M + c_{M-1} + c_{M-2})]$$  \hspace{1cm} (27)

$$f_3(c_M, c_{M-1}, ..., c_0) = \frac{\lambda}{\lambda + 3\Delta^2} [c_0 + \Delta a(c_M + c_{M-1} + c_{M-2})]^2$$  \hspace{1cm} (28)

or in general

$$v_k = -\frac{\Delta}{\lambda + k\Delta^2} [c_0 + \Delta a(c_M + c_{M-1} + ... + c_{M-k+1})]$$  \hspace{1cm} (29)

$$f_k(c_M, c_{M-1}, ..., c_0) = \frac{\lambda}{\lambda + k\Delta^2} [c_0 + \Delta a(c_M + c_{M-1} + ... + c_{M-k+1})]^2$$  \hspace{1cm} (30)

In this development, it is important to remember that $c_{M-1}$ of stage $k$ becomes the new $c_M$ of stage $(k - 1)$, and $c_{M-2}$ of stage $k$ becomes the new $c_{M-1}$ of stage $(k - 2)$, and so on. In sequence, each of the original state variables for the $k$-stage process become the final state variable $c_M$ for some succeeding stage. As a consequence of the decision at each stage, one new state variable is created, and the last one dropped. In the $k$-stage process after the passage of $(M + 1)$ stages, the original set of state variables $(c_M, c_{M-1}, ..., c_0)$ is completely replaced by a new set.

13. Alternate Approach to Time Lag Control

Let us consider an alternate approach to the same time delay problem with linear differential difference equations given in Section 12.
Let us normalize Eq. (1) of Section 12 so it can be written as follows:

\[
\frac{du}{dt} = cu(t - 1) + g(t), \quad t > 1
\]  

(1)

where \( u(t) \) exists over \( 0 \leq t \leq 1 \).

The historical condition is specified by

\[ u(t) = h(t); \quad 0 \leq t \leq 1 \]  

(2)

The variable \( g(t) \) is constrained by

\[ |g(t)| \leq k_1, \quad \text{for} \quad t > 1 \]  

(3)

\[ \int_1^T g^2(t) \, dt \leq k_2 \]  

(4)

Considering again the terminal control problem, the object of control is to minimize \( J(g) \)

\[ J(g) = \Phi(u(T)) + \lambda \int_1^T g^2(t) \, dt \]  

(5)

The correspondence between this formulation and the previous section is

\[ u(t) \sim x(t), \quad c \sim a, \quad g(t) \sim v(t) \]  

(6)

\[ \int_1^T g^2(t) \, dt \sim \int_0^T v^2(t) \, dt, \quad k_2 \sim V, \quad \Phi(u(T)) \sim x^2(T) \]

The only difference is that in Section 12 \( v(t) \) is not constrained as \( g(t) \) is in this section by Eq. (3).

The usual approach is to define a functional \( f(h(t), T) \):

\[ f(h(t), T) = \text{Min}_{g(t)} \left[ \Phi(u(T)) + \lambda \int_1^T g^2(t) \, dt \right] \]  

(7)

The state of the system being specified by a function, rather than by discrete variables raises a great number of difficulties in dynamic programming. Up to this point, except for Chapter 5, Section 31, we have considered only problems where the state of this system has been given by one or more discrete variables. The most desirable situation being only one variable. Where a distribution function is used to characterize the state of the system, it is extremely difficult to state unequivocally how the system passes from state to state. See Chapter 5, Section 31 for a discussion.
The approach given below circumvents the functional aspect of the state characterization by reducing for this linear case the problem to one employing only one variable as the state variable. The solution to Eq. (1) with conditions (2)–(4) is given by

\[ u(t) = h(1) K(t-1) + c \int_0^1 K(t - t_1 - 1) h(t_1) dt_1 + \int_1^T K(t - t_1) g(t_1) dt_1 \]

where the kernel \( K(t) \) is given by

\[ K'(t) = cK(t - 1), t > 1; \quad K(t) = 1, \quad 0 < t < 1; \quad K(t) = 0, \quad t < 0 \]

This solution was developed in reference [6].

We observe that the solution consists of a linear combination of the effect due to control; namely,

\[ h(1) K(t-1) + c \int_0^1 K(t - t_1 - 1) h(t_1) dt_1 \]

and the effect due to the historical condition; namely,

\[ h(1) K(t-1) + c \int_0^1 K(t - t_1 - 1) h(t_1) dt_1 \]

We may express \( u(t) \) as a function of the terminal time \( T \).

\[ \Phi(u(T)) = \Phi \left[ h(1) K(T-1) + c \int_0^1 K(t - t_1 - 1) h(t_1) dt_1 \right. \]

\[ \quad \left. + \int_1^T K(t - t_1) g(t_1) dt_1 \right] \]

or

\[ \Phi(u(T)) = \Phi \left[ b + \int_1^T K(T - t_1) g(t_1) dt_1 \right] \]

where

\[ b = h(1) K(T-1) + c \int_0^1 K(T - t_1 - 1) dt_1 \]

The \( b \) term may be considered to be the state of the system at time \( T \) if no control is exerted for \( t \geq 1 \). This is the condition for \( g(t) = 0 \) for \( t \geq 1 \).
Referring to the usual approach given in Eq. (7), we define a function \( f(b, T) \) as follows:

\[
f(b, T) = \min_{\varphi(t)} \left[ \varphi(b + \int_{1}^{T} K(T - t_1) g(t_1) dt_1 + \lambda \int_{1}^{T} g^2(t_1) dt_1 \right] \tag{13}
\]

where \( b \) is defined by Eq. (12) and \( T \) is the terminal time.

Let us break the time interval \((1, T)\) into two parts \((1, 1 + \Delta)\) and \((1 + \Delta, T)\) where \( \Delta \) is a small discrete time interval:

\[
f(b, T) = \min_{\varphi(t)} \left[ \varphi(b + \int_{1}^{1+\Delta} K(T - t_1) g(t_1) dt_1 + \int_{1+\Delta}^{T} K(T - t_1) g(t_1) dt_1 \right] + \lambda \int_{1}^{1+\Delta} g^2(t_1) dt_1 + \lambda \int_{1+\Delta}^{T} g^2(t_1) dt_1 \tag{14}
\]

Now, by definition of \( f(b, T) \) in Eq. (13),

\[
f(b_1, T - \Delta) = \min_{\varphi(t_1)} \left[ \varphi(b_1 + \int_{1}^{T} K(T - t_1) g(t_1) dt_1 + \lambda \int_{1+\Delta}^{T} g^2(t_1) dt_1 \right] \tag{15}
\]

If

\[
b_1 = b + \int_{1}^{1+\Delta} K(T - t_1) g(t_1) dt_1 \tag{16}
\]

then

\[
f(b + \int_{1}^{1+\Delta} K(T - t_1) g(t_1) dt_1, T - \Delta) = \\
\min_{\varphi(t)} \left[ \varphi(b + \int_{1}^{1+\Delta} K(T - t_1) g(t_1) dt_1 + \int_{1+\Delta}^{T} K(T - t_1) g(t_1) dt_1 \right] + \lambda \int_{1+\Delta}^{T} g^2(t_1) dt_1 \tag{17}
\]

Substituting Eq. (17) into Eq. (14) and using the Principle of Optimality over the span \((1, 1 + \Delta)\), we have

\[
f(b, T) = \min_{\varphi(t)} \left[ \lambda \int_{1}^{1+\Delta} g^2(t_1) dt_1 + f(b + \int_{1}^{1+\Delta} K(T - t_1) g(t_1) dt_1, T - \Delta) \right] \tag{18}
\]

For small \( \Delta \), this may be written

\[
f(b, T) = \min_{\varphi(t)} \left[ \lambda g^2(1) \Delta + f(b + K(T - 1) g(1) \Delta, T - \Delta) \right] \tag{19}
\]
The initial condition is

$$f(b, 0) = \Phi(b)$$

We observe in this formulation that by definition \( b \) is not a function of \( g(t_1) \). We also observe that \( b \) is a number. At the end of time period \( 1 \) to \( (1 + \Delta) \), or to put it another way, at the beginning of time period \( (1 + \Delta) \), the new value of \( b \) is

$$b + \int_{1}^{1+\Delta} K(T - t_1) g(t_1) \, dt_1$$

which is also a number. It also is not dependent on the choice of \( g(t) \) during the succeeding time interval \( (1 + \Delta) \) to \( (1 + 2\Delta) \). In this formulation, we have a definitive way to describe over the time period \( 1 \) to \( (1 + \Delta) \) the change in state from \( b \) to \( b + \int_{1}^{1+\Delta} K(T - t_1) g(t_1) \, dt_1 \), and of the duration of the process from \( T \) to \( (T' - \Delta) \).

The crux of the reduction of the state of a system from a functional given by \( f(h(t), T) \) in Eq. (7), where a distribution function describes the state of the system, to numbers \((b, T)\) describing the state of the system is the linearity of the solution of the differential difference equation. The numbers \( b \) and \( T \) embody all the information required to describe the transition of the state of the system from \( 1 \) to \( (1 + \Delta) \). The linear solution consists of a linear combination of the "noncontrol" term plus a control term.

The alternate approach differs from the previous discussion of the control problem with time lags in that the alternate approach requires a knowledge of the analytical solution of Eq. (1). From the analytical solution is built the optimal solution subject to the controls criteria and constraints. In the discrete formulation, step-by-step difference equations are solved without the knowledge of the formal analytical solution of Eq. (1). The finding of the analytical solution following the method given in this section may be an extremely difficult job. The finite difference approximation method reduces the problem to solving the controls problem directly, rather than generating analytical solutions to the original equation and then optimizing. The analytical solution approach has the advantage that only two state variables \( b \) and \( T \) are required to describe the system, while the previous method required a number of state variables.

We should note that this alternate procedure is the same technique discussed in the reduction of dimensionality in Chapter 5, Sections
14. Optimal Design of Control Systems

Fang and Higgins consider the optimal design of control systems using dynamic programming [26].

In Fig. 3 we have a typical closed loop feedback control system. Here, the s plane transfer function $G_2(s)$ is known as well as the input $R(s)$. The object of the design is to determine the transfer functions $G_1(s)$ and $H(s)$.

![Fig. 3](image)

If we write the ratio of the output $X(s)$ to input $R(s)$, we see that for Fig. 3

$$\frac{X(s)}{R(s)} = \frac{G_1(s) G_2(s)}{1 + G_1(s) G_2(s) H(s)}$$

and that for Fig. 4

$$\frac{X(s)}{R(s)} = \frac{R(s) + V(s)}{R(s)} G_2(s)$$
Equating the two transfer functions, we have

\[
\frac{G_1(s)}{1 + G_1(s) G_2(s) H(s)} = \frac{R(s) + V(s)}{R(s)} \tag{3}
\]

If \( G_2(s) \), \( R(s) \), and \( V(s) \) are specified, Eq. (3) yields a relationship between \( G_1(s) \) and \( H(s) \). This means that if a control problem corresponding to the block diagram in Fig. 3 is stated so that a control vector \( V(s) \) can be manipulated to meet the objective, then Fig. 4 can effectively replace Fig. 3. In other words, the transfer function of Eq. (2) replaces the transfer function of Eq. (1). Here, the \( V(s) \) or its counterpart, \( v(t) \) in the \( t \) plane, is manipulated in time to achieve the optimization.

The design of many control systems is pointed toward minimizing a performance index such as the integral of the error squared

\[
J(v) = \int_0^T [r(t) - x(t)]^2 dt \tag{4}
\]

where \( r(t) \) is the input signal and \( x(t) \) is the output signal. By means of dynamic programming, we may readily handle problems with performance indices such as Eq. (4).

Fang and Higgins consider the following performance index:

\[
J(v) = \int_0^T D'[r(t) - x(t)]^2 + A[r'(t) - x'(t)]^2 \] dt \tag{5}

where

\[
r(t) = \text{the input signal} \tag{6}
\]

\[
r'(t) = \frac{dr}{dt} \tag{7}
\]

\[
x(t) = \text{the output signal} \tag{8}
\]

\[
x'(t) = \frac{dx}{dt} = y \tag{9}
\]

\[
D' = \text{a weighting factor, where } D > 1 \tag{10}
\]

\[
A = \text{a constant} \tag{11}
\]

This problem is of similar form to the smoothing problem discussed in Chapter 4, Section 23.

Referring to Fig. 4, we consider \( G_2(s) \) to be the transfer function or the describing function of the equation

\[
x''(t) + G(x', x) = B(r + v) \tag{12}
\]
where
\[ B = \text{a constant} \]
\[ G(x', x) = \text{a known function, linear or nonlinear} \]
\[ v(t) = \text{the manipulated or control variable} \]

The \( v(t) \) is restricted by
\[ |v(t)| \leq V_1 \]

where
\[ V_1 = \text{a known constant} \]

The objective is to minimize \( J(v) \) in Eq. (5) subject to Eqs. (12)–(14), (16), and (17) by manipulation of \( v(t) \).

As before, we break the time interval \((0, T)\) into \( N \) equal increments of \( \Delta \) duration so \( N\Delta = T \). We count time forward so \( k = 0 \) refers to the initial time and \( k = N \) refers to the final time. We write \( x_k = x(k\Delta), r_k = r(k\Delta), v_k = v(k\Delta), \) etc.

The discrete form of the performance index is given by
\[ J_N(v_k) = \Delta \sum_{k=0}^{N-1} D^{kd}((r_k - x_k)^2 + A(r_k' - x_k')^2) \]  
\[ J_N(v_k) = \Delta[(r_0 - x_0)^2 + A(r_0' - x_0')^2] + \Delta \sum_{k=1}^{N-1} D^{kd}((r_k - x_k)^2 + A(r_k' - x_k')^2) \]

If we approximate \( dx/dt \) by
\[ \frac{x_k - x_{k-1}}{\Delta} = y_{k-1}, \quad k = 1, 2, \ldots, N \]

we may approximate Eq. (12) by
\[ y_k' = y_{k-1}' + \Delta[B(r_{k-1} + v_{k-1}) - G(y_{k-1}, x_{k-1})] \]

We now define
\[ J_N(x_0, y_0, r_0, v_0) = \text{Min } J_N(v_k) = \text{the minimum of the performance index over the } N \text{ stages starting in the initial state } (x_0, y_0, r_0, v_0), \text{ and pursuing an optimal policy} \]
Using the Principle of Optimality, we have

$$\begin{align*}
    f_N(x_0, y_0, r_0, r'_0) &= \min_{{v_0}} \left[ A((r_0 - x_0)^2 + A(r'_0 - y_0)^2) \right] \\
    &+ D^2 f_{N-1}(x_0 + \Delta y_0, y'_0 + \Delta(B(r_0 + v_0) - G(x_0, y_0)), r_1, r'_1) \\
    \text{and as boundary condition} \\
    f_0(x_0, y_0, r_0, r'_0) &= 0 
\end{align*}$$

and as boundary condition

$$f_0(x_0, y_0, r_0, r'_0) = 0$$

15. Numerical Example

Referring to the previous section, we evaluate Eq. (23) given the following data:

$$\begin{align*}
    x'' + ax' + bx &= B(1 + v) \\
    r_k &= 1, \quad k = 0, 1, 2, \ldots, N \\
    r'_k &= 0, \quad k = 0, 1, 2, \ldots, N \\
    x_0 &= 0, \quad k = 0, 1, 2, \ldots, N \\
    x'_0 &= y'_0 = 0, \\
    a &= 2 \\
    b &= 10 \\
    B &= 10 \\
    \Delta &= 0.1 \text{ sec} \\
    D &= 1.2 \\
    A &= 0.5 \\
    | v(t) | &\leq V_1 = 1.5
\end{align*}$$

Equation (1) here corresponds to Eq. (12) of Section 14. It supplies the data for the element $G_2(s)$ in Fig. 4.

Equations (2) and (3) describe the input to the system for all $k$. Equation (2) is the familiar unit step input. The initial conditions for $k = 0$ are given by Eqs. (2)–(5). The data equations (6)–(8) evaluate the constants in Eq. (1). The data equations (9)–(12) refer to Eq. (23) of the previous section.

The numerical solutions of Eq. (23) of Section 14 are listed in Table 1: $x_k, x'_k, v_k$, and $f_k$. Figures 5 and 6 show plots of the corresponding $x(t)$ and $v(t)$ as functions of time. In Fig. 5 also is shown the response
when no control is exerted; that is, $v(t) = 0$. For the process controlled by dynamic programming, the system reaches steady state in about 2.0 seconds. The uncontrolled process, on the other hand, is still hunting after 4.0 seconds. It exhibits overshoot and oscillation around the steady-state value. The dynamic programming control eliminates the overshooting and the oscillations.
### TABLE 1

**Evaluation of Response, Forcing Function, and Performance Index**

<table>
<thead>
<tr>
<th>$k$</th>
<th>$x_k$</th>
<th>$x'$</th>
<th>$v_k$</th>
<th>$f_k$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.0000</td>
<td>0.0000</td>
<td>1.3228</td>
<td>0.0000</td>
</tr>
<tr>
<td>1</td>
<td>0.0000</td>
<td>2.3228</td>
<td>-1.0760</td>
<td>0.1000</td>
</tr>
<tr>
<td>2</td>
<td>0.2323</td>
<td>1.7820</td>
<td>-0.8257</td>
<td>0.2200</td>
</tr>
<tr>
<td>3</td>
<td>0.4105</td>
<td>1.3676</td>
<td>-0.6343</td>
<td>0.3605</td>
</tr>
<tr>
<td>4</td>
<td>0.5473</td>
<td>1.0490</td>
<td>-0.4869</td>
<td>0.5170</td>
</tr>
<tr>
<td>5</td>
<td>0.6522</td>
<td>0.8050</td>
<td>-0.3744</td>
<td>0.6813</td>
</tr>
<tr>
<td>6</td>
<td>0.7327</td>
<td>0.6174</td>
<td>-0.2878</td>
<td>0.8432</td>
</tr>
<tr>
<td>7</td>
<td>0.7944</td>
<td>0.4734</td>
<td>-0.2210</td>
<td>0.9963</td>
</tr>
<tr>
<td>8</td>
<td>0.8417</td>
<td>0.3633</td>
<td>-0.1709</td>
<td>1.1318</td>
</tr>
<tr>
<td>9</td>
<td>0.8780</td>
<td>0.2780</td>
<td>-0.1315</td>
<td>1.2460</td>
</tr>
<tr>
<td>10</td>
<td>0.9058</td>
<td>0.2129</td>
<td>-0.1017</td>
<td>1.3390</td>
</tr>
<tr>
<td>11</td>
<td>0.9271</td>
<td>0.1628</td>
<td>-0.0788</td>
<td>1.4115</td>
</tr>
<tr>
<td>12</td>
<td>0.9434</td>
<td>0.1243</td>
<td>-0.0611</td>
<td>1.4668</td>
</tr>
<tr>
<td>13</td>
<td>0.9558</td>
<td>0.0949</td>
<td>-0.0480</td>
<td>1.5080</td>
</tr>
<tr>
<td>14</td>
<td>0.9653</td>
<td>0.0721</td>
<td>-0.0377</td>
<td>1.5380</td>
</tr>
<tr>
<td>15</td>
<td>0.9725</td>
<td>0.0547</td>
<td>-0.0302</td>
<td>1.5600</td>
</tr>
<tr>
<td>16</td>
<td>0.9780</td>
<td>0.0411</td>
<td>-0.0243</td>
<td>1.5760</td>
</tr>
<tr>
<td>17</td>
<td>0.9821</td>
<td>0.0306</td>
<td>-0.0199</td>
<td>1.5875</td>
</tr>
<tr>
<td>18</td>
<td>0.9852</td>
<td>0.0225</td>
<td>-0.0166</td>
<td>1.5956</td>
</tr>
<tr>
<td>19</td>
<td>0.9875</td>
<td>0.0162</td>
<td>-0.0140</td>
<td>1.6016</td>
</tr>
<tr>
<td>20</td>
<td>0.9891</td>
<td>0.0115</td>
<td>1.6058</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>0.9903</td>
<td></td>
<td>1.6087</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td></td>
<td></td>
<td>1.6104</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td></td>
<td></td>
<td>1.6118</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td></td>
<td></td>
<td>1.6128</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td></td>
<td></td>
<td>1.6136</td>
<td></td>
</tr>
</tbody>
</table>

**REFERENCES**


PROBLEMS

1. Minimize over \( y(t) \)

\[ J(y) = \int_0^T [1 - e^{-ay}] \, dt \]

subject to

\( a) \frac{dx}{dt} = \frac{x}{y^2} \quad \text{and} \quad (c) \quad T = 10 \)

\( b) \quad x(0) = 5 \)

2. Minimize the final value of \( [1 - e^{ay}] \), namely, \( [1 - e^{-a(T)y(T)}] \) over \( y(t) \)

where

\( a) \frac{dx}{dt} = \frac{x}{y^2} \quad \text{and} \quad (c) \quad \int_0^T xy \, dt \leq 20 \)

\( b) \quad x(0) = 5 \)

\( c) \quad \int_0^T xy \, dt \leq 20 \)

\( d) \quad T = 10 \)
PROBLEMS

3. We desire to minimize the performance index $J(x) = \int_0^T x^2 \, dt$ for a second-order system with constant coefficients characterized by

$$\frac{dx^2}{dt^2} + a \frac{dx}{dt} + bx = f(t)$$

with the initial conditions

$$x_0 = c_1 \quad \text{and} \quad x'_0 = c_2$$

and the constraint

$$|f(t)| \leq A = \text{constant}$$

On defining

$$F_N(c_1, c_2) = \min_{f_k} \sum_{k=0}^{N-1} x^2 \Delta$$

show that the functional equations are

$$F_{k+1}(c_1, c_2) = \min f_k \left[ c_1^2 \Delta + F_k(c_1 + c_2 \Delta, c_2 + (f_0 - ac_2 - bc_1) \Delta) \right]$$

$$F_1(c_1, c_2) = c_1^2 \Delta$$

(Schlager, K. J., and Higgins, T. J. [37].)

4. Taking $a = 1, b = 2, A = 3, \Delta = 1, c_1 = 1, c_2 = 2$, in Problem 3 evaluate $F_N(c_1, c_2)$ and the policy $f_k$ for a 10-stage process.

5. Suppose it is desired to control a system to drive it from its initial conditions to quiescence in minimum time. The system equations are

$$\frac{dx^3}{dt^3} + a_1 \frac{d^2 x}{dt^2} + a_2 \frac{dx}{dt} + a_3 x = f(t)$$

where $a_1, a_2$, and $a_3$ are constants. The forcing function $f(t)$ is constrained to take the values of $+b$ or $-b$. The initial conditions are given by

$$x_0 = c_1, \quad x'_0 = c_2, \quad x''_0 = c_3$$

Show that the optimum time is determined by

$$F_{k+1}(c_1, c_2, c_3) = \Delta + \min_{f_k} F_k(c_1 + c_2 \Delta, c_2 + c_3 \Delta, c_3 + (f_0 - a_1c_3 - a_2c_2 - a_3c_1) \Delta)$$

6. We desire to minimize the performance index

$$J = \int_0^T \left[ u^2 + \left( \frac{du}{dt} - 1 \right)^2 \right] \, dt$$
for a system described by
\[
\frac{d^2u}{dt^2} + (u^2 - 1) \frac{du}{dt} + u = f(t)
\]
where \(f(t)\), the control variable, satisfies
\[-1 \leq f(t) \leq 1\]
The initial conditions at \(t = 0\) are
\[u_0 = c_1 = 0, \quad u'_0 = c_2 = 1\]
For \(T = 12\) and \(\Delta = 1\), check the validity of the accompanying tabulation.

<table>
<thead>
<tr>
<th>Stages</th>
<th>State variable</th>
<th>Decision function</th>
<th>Criterion function</th>
</tr>
</thead>
<tbody>
<tr>
<td>(k)</td>
<td>(u_k)</td>
<td>(y_k)</td>
<td>(f_k)</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>-1</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

7. We desire to determine the function \(g(t)\) which minimizes
\[
J(g) = \int_0^T [u^2 + g^2] dt
\]
where
\[
\frac{du}{dt} = u' = g, \quad 0 \leq t \leq T
\]
\[u(0) = c\]
On defining \(f(c, T) = \text{Min} \ J(g)\) show that
(a) \(f_T = \text{Min}_w \left( c^2 + w^2 + w \frac{\partial f}{\partial c} \right)\), \(g(0) = w\)
and \(f(c, 0) = 0\)
(b) \(w_{\text{Min}} = -\frac{1}{2} f_c\)
(c) \(f_T = c^2 - \frac{1}{4} f_c^2\)
Assuming that \( f(c, T) = \varphi(T)c^2 \), where \( \varphi(T) \) is a function to be determined, show that \( \varphi(T) \) satisfies

\[(d) \quad \frac{d\varphi}{dT} = 1 - \varphi^2 \quad \text{with} \quad \varphi(0) = 0\]

\[(e) \quad \varphi(T) = \tanh T \quad \text{and} \quad f(c, T) = c^2 \tanh T\]

\[(f) \quad \omega_{\text{Min}} = -c \tanh T\]

For a short process, that is where \( w = 0 \), show that

\[(g) \quad \int f(g) = c^2 T\]

For a long process, where \( w = -c \), show that

\[(h) \quad \int f(g) = c^2 (1 - e^{-2T})\]

(Kalaba, R., Computational consideration for some deterministic and adaptive control processes. The RAND Corporation, P-2210 (January 17, 1961).)

8. Using the calculus of variations, find the function \( u(t) \), which minimizes

\[ J(u) = \int_0^T [u^2 + (u')^2] \, dt \]

subject to the boundary conditions

\[ u(0) = 0, \quad u'(T) = 0 \]

Note that this problem and the previous one are equivalent

Answer: \[ u(t) = \frac{c \cosh (t - T)}{\cosh T}; \quad u'(t) = \frac{c \sinh (t - T)}{\cosh T} \]

If \( f(c, T) = \operatorname{Min}_{g} J(g) \) where

\[ J(g) = \int_0^T [(u^2 + g^2) \, dt + u^2(T)] \, dt, \quad u' = g, \quad 0 \leq t \leq T \]

\[ u(0) = c \]

show that

\[ f(c, T) = c^2 \]

9. We desire to minimize \( x(T) \) subject to

\[ \frac{dx}{dt} = ax(t) + v(t), \quad x(0) = c_0 \]
6. CONTROL PROBLEMS

where

\[ \int_0^T v^2(t) \, dt \leq V \]

(a) Set up the functional equations.
(b) Compare the solution to the retarded control problem in Section 12.
What is the price we pay for retarded control?

10. For the time lag problem in Section 12, minimize

\[ \int_0^T x(t) \, dt \]

subject to

\[ \frac{dx(t)}{dt} = ax(t - T_0) + v(t) \]

where

\[ \int_0^T v^2(t) \, dt \leq V \]

11. Show that the solution of the retarded control problem in Section 12 may be expressed for \( D = 1 \) by

\[ f_k(S) = \frac{\lambda[\Sigma_{j=0}^M \alpha_j^k c_j]^2}{\lambda + Q_k} \quad \text{ and } \quad v_k(S) = \frac{-[\Sigma_{j=0}^M \alpha_j^k c_j] \alpha_0^{k-1}}{\lambda + Q_k} \]

where

\[ \alpha_j^k = \begin{cases} a \alpha_0^{k-1}, & j = M \\ \alpha_j^{k-1}, & 1 \leq j \leq M - 1 \\ \alpha_1^{k-1} + \alpha_0^{k-1}, & j = 0 \end{cases} \]

\( \alpha_j = \begin{cases} a, & j = M \\ 0, & 1 \leq j \leq M - 1 \\ 1, & j = 0 \end{cases} \)

\( Q_k = Q_{k-1} + (\alpha_0^{k-1})^2, \quad Q_1 = 1 \)

(Kramer, J. D. R. [32].)

12. Suppose we desire to minimize

\[ \lambda \sum_{j=1}^N v_j^2 + x_0^2 \]

where \( x_n \) and \( v_n \) are connected by the general linear difference equation

\[ x_{n-1} = a_0 x_n + a_1 x_{n+1} + \ldots + a_M x_{n+M} + v_n \]
We define the following:

\[ S = \text{initial state} = (C_M, C_{M-1}, \ldots, C_0), \text{ vector} \]
\[ S_k = \text{state of system at } k\text{th interval} = (x_{k+M}, x_{k+M-1}, \ldots, x_k), \text{ vector} \]

\[
P = \begin{pmatrix}
0 & 0 & \ldots & 0 & a_M \\
1 & 0 & \ldots & 0 & a_{M-1} \\
0 & 1 & \ldots & 0 & a_{M-2} \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \ldots & 0 & 1 \cdot a_0
\end{pmatrix}, \quad M \times M \text{ matrix}
\]

\[ V_j = (0 \ 0 \ \ldots \ v_j), \text{ a } 1 \times M \text{ vector} \]
\[ A_k = \text{last column of the matrix } P^k, \text{ a vector} \]
\[ H_{-k} = \text{lower right-hand element of the matrix } P^{k-1}, \text{ a vector} \]

We note that the final position \(x_0\) of a system which is in state \(S\) at time \(k\) may be expressed as

\[ x_0 = (S, A_k) + \sum_{j=1}^{k} v_j H_{-j} \]

The inner product term \((S, A_k) = B_k = \text{trend number}\). The trend number specifies the state of the system. A recursion relationship exists between trend numbers \(B_k\) and \(B_{k-1}\) and the control force \(v_k\), namely,

\[ B_{k-1} = B_k + v_k H_{-k} \]

Let us define

\[ f_k(B_k) = \min_{v_k} \left[ \lambda \sum_{j=1}^{k} v_j^2 + x_0^2 \right] \]

Show that

\[ f_k(B_k) = \min_{v_k} \left[ \lambda v_k^2 + f_{k-1}(B_k + v_k H_{-k}) \right]; \quad f_0(B_0) = B_0^2 \]

\[ f_k(B_k) = \frac{\lambda B_k^2}{\lambda + Q_k}, \quad v_k(B_k) = \frac{-B_k}{\lambda + Q_k} H_{-k} \]

where

\[ Q_k = \sum_{j=1}^{k} H_j^2 \]
Note that the differential difference equation may be written as \( S_{k-1} = S_k P \) from which it follows \( S_0 = SP^k + \sum_{j=1}^{k-1} V_j P^{j-1} \). Justify the definitions of \( A_k \) and \( H_{-k} \).

(Kramer J. D. R. [32].)

13. Given a linear differential equation

\[
L[x(t)] = v(t)
\]

where \( L[\cdot] \) is a linear differential operator. We desire to minimize

\[
J(v) = \lambda \int_0^T v^2(t) \, dt + x^2(0)
\]

This is a statement of the continuous version of Problem 12.

We note that the terminal position \( x(0) \) for a process beginning at time \( T \) is given by

\[
x(0) = B(T) + \int_0^T v(\tau) H(-\tau) \, d\tau
\]

where \( B \) and \( H \) are defined in the previous problem. This is the continuous analog to the discrete version in Problem 12. Here \( B(T) \) is the trend number.

If we define

\[
f(T, B) = \min_{v(t)} J(v)
\]

show that

\[
f(T, B) = \min_{v} [\lambda v^2 + f(T - \delta, B(T) + \delta v(T) H(-T))]\]

\[
f(0, B) = B^2
\]

where \( \delta \) is a small increment of time.

Show that the solution of the functional equations is given by

\[
2\lambda \psi(T) + \frac{\partial f(T, B)}{\partial B} H(-T) = 0
\]

\[
\lambda \psi^2(T) - \frac{\partial f(T, B)}{\partial T} + \frac{\partial f(T, B)}{\partial B} \psi(T) H(-T) = 0
\]

\[
f(0, B) = B^2
\]

or

\[
f_T + f_B^2 \frac{H_T(-T)}{4\lambda} = 0, \quad f(0, B) = B^2
\]
14. From the previous problem, taking
\[ f_T + f_B^2 \frac{H^2(-T)}{4\lambda} = 0, \quad f(0, B) = B^2 \]
and assuming
\[ f(T, B) = g(T)h(B) \]
show that
\[ f(T, B) = \frac{\lambda B^2(T)}{\lambda + \int_0^T H^2(-t) dt}, \quad v(T, B) = \frac{B(T) H(-T)}{\lambda + \int_0^T H^2(-t) dt} \]
Compare these solutions to those in Problem 12.

15. Given the matrix equation
\[ x'(t) = Ax(t) + Dm(t) + Hu(t) \]
show that
\[ x(t) = \phi(t - t_0) x(t_0) + \int_{t_0}^t \phi(t - \tau) Dm(\tau) d\tau + \int_{t_0}^t \phi(t - \tau) Hu(\tau) d\tau \]
where \( \phi(t) \) is the solution to
\[ x'(t) = Ax(t) \quad \text{and} \quad x(0) = I \]
where \( x(t) \) = state vector, \( n \times 1 \)
\( m(t) \) = manipulated vector, \( r \times 1 \)
\( u(t) \) = disturbance vector, \( s \times 1 \)
\( A \) = \( n \times n \) matrix
\( D \) = \( n \times r \) matrix
\( H \) = \( n \times s \) matrix

Show for time \( t = (k + 1)T \), where \( T \) is the sampling interval, that
\[ x((k + 1)T) = \phi(T) x(kT) + h(T) m(kT) + U(kT) \]
where
\[ h(T) = \int_{kT}^{(k+1)T} \phi((k + 1) T - \tau) D d\tau \]
\[ U(kT) = \int_{kT}^{(k+1)T} \phi((k + 1) T - \tau) Hu(\tau) d\tau \]
(Tou, J. [38].)
16. Given

\[ J_N = \sum_{k=1}^{N} x^T(k) Q x(k) \]

and

\[ f_{N-j}[x(j)] = \min_{m(j)} \sum_{k=j+1}^{N} x^T(k) Q x(k), \quad f_0[x(0)] = 0. \]

Assuming \( f_{N-j}[x(j)] = x^T(j) P(N - j) x(j), \) show for \( m(j), \) a \( 1 \times 1 \) vector, that:

\[ f_{N-j}[x(j)] = \min_{m(j)} \{ \phi(T) x(j) + h(T) m(j) \}^T \times S(N - j + 1) [\phi(T) x(j) + h(T) m(j)] \]

where

\[ S(N - j + 1) = Q + P(N - j + 1) \]

and where \( Q \) and \( P(N - j) \) are \( n \times n \) symmetric matrices.

Using the definitions in Problem 15 show that

(a) \( 2 h^T(T) S(N - j + 1) h(T) m(j) + h^T(T) S(N - j + 1) \phi(T) x(j) \]
\[ + x^T(j) \phi^T(T) S(N - j + 1) h(T) = 0 \]

(b) \( m^0(j) = B^T(N - j) x(j) = \) optimal control vector

where

\[ B^T(N - j) = \frac{-h^T(T) [Q + P(N - j + 1)] \phi(T)}{h^T(T) [Q + P(N - j + 1)] h(T)} \]

(c) \( P(N - j) = [\phi(T) + h(T) B^T(N - j)]^T [Q + P(N - j + 1)] \]
\[ \times [\phi(T) + h(T) B^T(N - j)] \]

(d) Show that the control sequence is found by starting with \( P(0) = 0 \) and computing \( B^T(1), P(1), B^T(2), P(2), \) etc.

(Tou, J. [38].)

17. Given

\[ J_N = \sum_{k=1}^{N} x^T(k) Q x(k) + \rho m^2(k - 1) \]

and

\[ f_{N-j}[x(j)] = \min_{m(j)} \sum_{k=j+1}^{N} [x^T(k) x(k) + \rho m^2(k - 1)] \]
Using the definitions in Problem 15 show for \( m(j) \), a \( 1 \times 1 \) vector that
\[
m^0(j) = B^T(N - j) x(j)
\]
\[
B^T(k) = \frac{-h^T(T)[Q + P(k - 1)] \phi(T)}{h^T(T)[Q + P(k - 1)] h(T) + \rho}
\]
\[
P(k) = [\phi(T) + h(T) B^T(k)]^T [Q + P(k - 1)] [\phi(T) + h(T) B^T(k)] + \rho B(k) B^T(k)
\]

18. The control objective is to minimize
\[
J_N = \sum_{k=1}^{N} [x^2(k) + \rho m^2(k - 1)]
\]
and the system behavior is governed by
\[
x' = -ax + bm
\]
Using the definitions in Problem 15 show that
(a) \( \phi(T) = e^{-aT} \)
(b) \( h(T) = \frac{b}{a} (1 - e^{-aT}) \)
(c) \( x(k + 1) = e^{-aT} x(k) + \frac{b}{a} (1 - e^{-aT}) m(k) \)
(d) \( B^T(1) = \frac{-ab(1 - e^{-aT}) e^{-aT}}{b^2(1 - e^{-aT})^2 + a^2 \rho} \) for \( k = 1 \) (use the fact that \( Q = 1 \) and \( P(0) = 0 \))
(e) \( P(1) = \frac{a^2 \rho e^{-2aT}}{a^2 \rho + b^2(1 - e^{-aT})^2} \)
(f) \( B^T(2) = \frac{-ae^{-aT}[a^2 \rho (1 + e^{-2aT}) + b^2(1 - e^{-aT})^2]}{b(1 - e^{-aT}) [a^2 \rho (1 + e^{-2aT}) + b^2(1 - e^{-aT})^2] + \rho [a^2 \rho + b^2(1 - e^{-aT})^2]} \)

(Tou, J. [38]).

19. Suppose we desire to minimize
\[
J_N = \sum_{k=1}^{N} (x_k - r_k)^2
\]
where \( r_k \) is a unit step function. The system is governed by
\[
x'' + x' = m
\]
Using the definitions in Problem 15 and the following substitutions
\[ v = \begin{bmatrix} x & y & r \end{bmatrix}^T, \quad d = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}^T \]
\[ y = x' \]
\[ A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \]
show that the control equation may be written as \[ v' = Av + dm \]

Check the validity of the following:

(a) \[ \phi(T) = \begin{bmatrix} 1 & 1 - e^{-T} & 0 \\ 0 & e^{-T} & 0 \\ 0 & 0 & 0 \end{bmatrix} \]

(b) \[ h(T) = \begin{bmatrix} T - 1 + e^{-T} \\ 1 - e^{-T} \\ 0 \end{bmatrix} \]

(c) \[ J_N = \sum_{k=1}^{N} v^T(k)Qv(k) \]
where
\[ Q = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix} \]

20. For the previous problem check the results that for \( p = 0, T = 1 \) second
\[ B^T(k) = -2.7183, \quad -1.7182, \quad +2.7183 \]
and the optimal policy is given by
\[ m^*(k) = -2.7183x(k) - 1.7182x'(k) + 2.7183r(k) \]
Check the following computations for minimizing \( \sum_{k=1}^{N} (x_k - r_k)^2 + \rho m^2(k - 1) \).

<table>
<thead>
<tr>
<th>( k )</th>
<th>( \rho = 1, \ T = 1 )</th>
<th>( B'(k) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-0.32403</td>
<td>-0.20482</td>
</tr>
<tr>
<td>2</td>
<td>-0.63106</td>
<td>-0.49394</td>
</tr>
<tr>
<td>3</td>
<td>-0.69648</td>
<td>-0.57780</td>
</tr>
<tr>
<td>4</td>
<td>-0.70106</td>
<td>-0.58861</td>
</tr>
<tr>
<td>5</td>
<td>-0.70068</td>
<td>-0.58912</td>
</tr>
<tr>
<td>6</td>
<td>-0.70064</td>
<td>-0.58910</td>
</tr>
<tr>
<td>7</td>
<td>-0.70068</td>
<td>-0.58913</td>
</tr>
<tr>
<td>8</td>
<td>-0.70070</td>
<td>-0.58914</td>
</tr>
<tr>
<td>9</td>
<td>-0.70070</td>
<td>-0.58915</td>
</tr>
</tbody>
</table>
21. If the performance index is a function of time through $Q$, namely, $Q(k)$, show that if we desire to minimize

\[ J_N = \sum x^T(k) Q(k) x(k) \]

for any arbitrary initial state $x(0)$ that

\[ f_N[x(0)] = \min_{m(0)} \sum_{k=1}^{N} x^T(k) Q(k) x(k) \]

\[ f_{N-j}[x(j)] = \min_{m(j)} \sum_{k=j+1}^{N} x^T(k) Q(k) x(k) \]

with $f_0[x(0)] = 0$. 

\[ \rho = 5, \ T = 1 \]

\[ \begin{array}{ccc}
k & B'(k) \\
1 & -0.07164 & -0.04528 & 0.07164 \\
2 & -0.19533 & -0.15391 & 0.19533 \\
3 & -0.28816 & -0.24600 & 0.28816 \\
4 & -0.33698 & -0.29749 & 0.33698 \\
5 & -0.35822 & -0.32071 & 0.35822 \\
6 & -0.36659 & -0.33009 & 0.36659 \\
7 & -0.36973 & -0.33365 & 0.36973 \\
8 & -0.37087 & -0.33496 & 0.37087 \\
9 & -0.37127 & -0.33543 & 0.37127 \\
\end{array} \]

\[ \rho = 1, \ T = 2 \]

\[ \begin{array}{ccc}
1 & -0.49600 & -0.42887 & 0.49600 \\
2 & -0.51015 & -0.46619 & 0.51015 \\
3 & -0.50941 & -0.46582 & 0.50941 \\
4 & -0.50946 & -0.46587 & 0.50946 \\
5 & -0.50946 & -0.46587 & 0.50946 \\
\end{array} \]

\[ \rho = 5, \ T = 2 \]

\[ \begin{array}{ccc}
1 & -0.18053 & -0.15610 & 0.18053 \\
2 & -0.28902 & -0.26914 & 0.28902 \\
3 & -0.30886 & -0.29121 & 0.30886 \\
4 & -0.31149 & -0.29423 & 0.31149 \\
5 & -0.31181 & -0.29459 & 0.31181 \\
6 & -0.31184 & -0.29463 & 0.31184 \\
7 & -0.31184 & -0.29464 & 0.31184 \\
8 & -0.31184 & -0.29464 & 0.31184 \\
\end{array} \]
Show using the Principle of Optimality and the definitions in Problem 15 for \( m(j) \), a \( 1 \times 1 \) vector, that
\[
f_{N-j}(x(j)) = \min_{m(j)} [x^T(j+1)Q(j+1)x(j+1) + f_{N-j+1}(x(j+1))]
\]
and that the optimal policy is
\[
m^0(j) = B^T(N - j)x(j)
\]
where
\[
B^T(N - j) = \frac{-h^T(j)[Q(j+1) + P(N - j + 1)]\phi(j)}{h^T(j)[Q(j+1) + P(N - j + 1)]h(j)}
\]
and
\[
P(N - j) = [\phi(j) + h(j)B^T(N - j)]^T[Q(j+1) + P(N - j + 1)]
\times [\phi(j) + h(j)B^T(N - j)]
\]
(Tou, J. [38]).

22. If \( m(j) \) is a \( 1 \times 1 \) vector and if we desire to minimize
\[
J_N = \sum_{k=1}^{N} [x^T(k)Q(k)x(k) + \rho m^2(k - 1)]
\]
show by the definitions in Problem 15 that the optimal control sequence is
\[
m^0(j) = B^T(N - j)x(j)
\]
\[
B^T(N - j) = \frac{-h(j)[Q(j+1) + P(N - j + 1)]\phi(j)}{h^T(j)[Q(j+1) + P(N - j + 1)]h(j) + \rho}
\]
\[
P(N - j) = [\phi(j) + h(j)B^T(N - j)]^T[Q(j+1) + P(N - j + 1)]
\times [\phi(j) + h(j)B^T(N - j)] + \rho B(N - j)B^T(N - j)
\]

23. Given
\[
x_N = x_{N-1} + y_{N-1}\tau - \frac{\tau^2}{2(1 + j_{N-1})} u,
\]
\[
y_N = y_{N-1} - \frac{\tau}{1 + j_{N-1}} u
\]
\[
u = \pm 1
\]
\[
j_N = \begin{cases} a & \text{with probability } p \\ -a & \text{with probability } 1 - p \end{cases}
\]
Define
\[
g(x, y) = \text{the expected minimum time to reach the origin } (0, 0) \text{ from the point } (x, y) \text{ following an optimal policy}
\]
Show that
\[ g(x, y) = \tau + \min_u [pg(x^+, y^+) + (1 - p)g(x^-, y^-)] \]
where
\[ x^+ = x + y\tau - \frac{\tau^2}{2(1 + a)} u, \quad y^+ = y - \frac{\tau}{1 + a} u \]
\[ x^- = x + y\tau - \frac{\tau^2}{2(1 - a)} u, \quad y^- = y - \frac{\tau}{1 - a} u \]

Since \( g(x, y) \) appears in both sides of the equation, discuss how this equation may be solved.

(Aoki, M., Dynamic programming and conditional response extrapolation. AIEE Workshop on Dynamic Programming, Boulder, Colorado, June 27, 1961.)

24. Given a set of differential equations in matrix form
\[ \frac{ds}{dt} = As + v \]
Define for a bang-bang control system
\[ f(s, t) = \min_v f(s + (As + v)A, t + A) \]
where \( v_1 \) takes on the values of either \((v_1)_{\text{Max}}\) or \(-(v_1)_{\text{Max}}\)
Letting \( A \to 0 \) show that
\[ - \frac{\partial f}{\partial t} = \min_v \{\nabla f \cdot (As + v)\} = (\nabla f \cdot As) + \min_v \{\nabla f \cdot v\} \]
Show that the ith component of the optimal control vector is
\[ v_i = - |v_i|_{\text{Max}} \text{sgn} \left( \frac{\partial f}{\partial s_i} \right) \]
(Aoki, M. Dynamic programming and conditional response extrapolation. AIEE Workshop on Dynamic Programming, Boulder, Colorado, June 27, 1961.)

25. Minimize \( J(u) \) where
\[ J(u) = \max_{0 \leq t \leq T} |1 - u| \]
subject to
(a) \( u'' + u = f \)
(b) \( u(0) = 0 \)
(c) \( u'(0) = 1 \)
(d) \( -1 \leq f \leq 1 \)
Let \( u' = v, \ v' = -u + f, u(0) = c_1, \ v(0) = c_2 \) and define

\[
 f_N(c_1, c_2) = \min_y \max_{\theta \leq k \leq N} |1 - u_k | 
\]

Show that

\[
 f_0(c_1, c_2) = |1 - c_1 |
\]

\[
 f_{N+1}(c_1, c_2) = \max [ |1 - c_1 |, \min_{-1 \leq f_0 \leq 1} f_N(c_1 + \Delta c_2, c_2 + \Delta (f_0 - c_1) ] 
\]

(Bellman, R. [1/1]).
7

Chemical Engineering
Control Problems

1. Introduction

In Chapter 6, we have shown the utility of dynamic programming in a variety of control problems. Much of the discussion is abstract to the chemical engineer, or uses equations such as Van der Pol's with which the chemical engineer is only slightly, if at all, familiar. In order to cast some of these problems in a more familiar light, we shall discuss some applications of dynamic programming to control problems in chemical engineering. It should be remembered, however, that chemical engineering control problems are basically no different from control problems in other fields such as servo, economics, and biology.

In the first sections of this chapter, we present a simple deterministic tankage control problem and then some variations of it. We next set up in Sections 6 and 7 the equations for a tubular chemical reactor and solve both a terminal control and averaging control problem for it. In contrast to previously discussed control problems, the manipulated variable, heat flux in this case, does not appear directly in the functional equations. The remainder of the chapter is devoted to the interesting work of Kalman, Lapidus, and Shapiro on the control of linear systems with a quadratic objective function. In Section 9, the basic equations for linearization around an equilibrium point are developed. An exposition on the performance index chosen is given in Section 10. Combining the results of Sections 9 and 10, we derive in detail in Section 11 the
control equations and the computational procedure. An example of the transient behavior of an absorber using this technique is presented in Sections 12 and 13 with numerical results. The results of Section 11 are rederived in Section 14 by an alternate method of Kalman. Finally, Kalman's method is reviewed in Section 15.

2. Tank Control Problem. Single Tank

a. Problem Statement

The problem we discuss in Fig. 1 falls under the general heading of scheduling. Here we have a tank, initially at level \( h \), which is required to deliver a desired effluent rate \( Q_\text{E}^* \) over a period of time. The feed rate \( Q_i \) is to be manipulated over time to meet the desired effluent rate.

\[\text{Fig. 1.}\]

The criterion of performance of this control system is to minimize \( \Sigma_{i=1}^{N}(Q_\text{E}^* - Q_\text{E})^2 \) over \( N \) stages of time, where \( Q_\text{E} \) is the actual effluent rate.

In the deterministic case, \( Q_\text{E}^* \), the desired effluent rate, is specified over time and does not change during the \( N \) time stages.

In the stochastic version, which is discussed in Chapter 9, Section 10, \( Q_\text{E}^* \), the estimated desired effluent rate, is perhaps an educated guess with the most recent information being more likely than the distant future. The uncertainty in the \( Q_\text{E}^* \) prognostication is one important probabilistic aspect. There are other stochastic elements, however, such as the uncertainties in pump efficiencies, orifice coefficients, and leakage.

In the adaptive version, discussed in Chapter 9, Section 10, the probability density function of \( (Q_\text{E}^* - Q_\text{E})^2 \) is re-evaluated periodically on the basis of new information to give new and hopefully better estimators. The principal difference between the stochastic version and
the adaptive version is that while both describe uncertainty models, the adaptive model takes advantage of past information to develop better prognosticators for the future.

b. Deterministic Case

The flow rate from a single tank is to be controlled so that the square of the difference between the desired effluent rate and the actual rate, summed up over $N$ stages of time, is to be a minimum. A material balance relationship must be satisfied over the $N$ stages of time:

$$\sum_{i=1}^{N} (Q_i) = \sum_{i=1}^{N} (Q_{E}) + A(h_i - h_N) \quad (1)$$

where $h_i$ is the head in the tank at time $i$ and $A$ is the cross sectional area. The instantaneous material balance at the $i$th stage is

$$Q_i = (Q_{E}) + A(h_i - h_{i-1}) \quad (2)$$

The discharge flow rate is proportional to the square root of the head:

$$Q_{E} = K_E \sqrt{h_i} \quad (3)$$

The optimization problem is to minimize

$$\sum_{i=1}^{N} (Q_{E}^*-Q_{E})^2 \quad (4)$$

where $Q_{E}^*$ is the desired discharge rate. This is specified as a function of the time [see Eq. (8) below]. The optimization is executed by varying $(Q_i)$ over the $N$ stages of time. The system is constrained by

$$h_{Min} \leq h_i \leq h_{Max} \quad (5)$$

$$(Q_{I})_{Min} \leq (Q_i) \leq (Q_{I})_{Max} \quad (6)$$

$$(Q_{E})_{Min} \leq (Q_{E}) \leq (Q_{E})_{Max} \quad (7)$$

$$(Q_{E}^*) = C_i, \quad i = 1, 2, ..., N \quad (8)$$

We now define

$$f_{X}(h) = \text{the value of the squared error between the desired effluent rate and the actual delivered effluent rate, summed over the } N \text{ remaining stages of time, beginning in the state } h, \text{ subject to the constraints (1)-(3) and (5)-(8), and using an optimal policy}$$
Using this definition we write

\[ f_N(h) = \min_{(Q_i)} \sum_{i=1}^{N} (Q_i^* - Q_i)^2 \]  \hspace{1cm} (9)

Using the Principle of Optimality, this may be written as

\[ f_N(h) = \min_{(Q_i)} \left[ (Q_N^* - Q_N)^2 + \sum_{i=1}^{N-1} (Q_i^* - Q_i)^2 \right] \]  \hspace{1cm} (10)

whence

\[ f_N(h) = \min_{(Q_i)} [(Q_N^* - Q_N)^2 + f_{N-1}(h')], \quad N \geq 2 \]  \hspace{1cm} (11)

and

\[ f_1(h) = \min_{(Q_i)} (Q_1^* - Q_1)^2 \]  \hspace{1cm} (12)

Here \( h' \), the new value of \( h \) at the beginning of stage \( (N - 1) \), is determined by (2) and (3). Equations (11) and (12) may be solved by the method described in Chapter 5, Section 4.

c. Discussion

In the formulation and solution discussed, it has been tacitly assumed that the \( Q_i \) rate can be adjusted at each stage to change the level by the required amount \( \Delta h \). Such a condition is closely approximated when the transient responses of the feed and effluent rates are small compared to the sampling period (time between stages). If the transient response of the feed and effluent rates are slow compared to the sampling period, then the level in the tank may never reach the required level. Under these circumstances, the treatment above should be modified to take into account the transient response.

If the effluent flow rate were controlled by a pump, Eq. (3) might be replaced by other expressions such as \( Q_E = Kh \). In either case, the form of the Eq. (3) or its counterpart does not affect the validity of Eqs. (11) and (12).

The method presented here has application in a number of useful areas. The continuous blending of various stocks from a multi-tank system is one example. Here the desired flow rate from each stock tank may vary as a function of time due to making a continuous variety of blended products of different quality. The desired flow rate may change due to a sampling and analysis of the blended products which require quality compensations. The desired flow rate may vary as a function of time due to starting up or shutting down units.
3. Two-Tank Cascade System

We desire to control the effluent $Q_E$ from the lower tank in Fig. 2 by manipulating the feed rate $Q_I$ to the upper tank. The object of control is to minimize $\sum_{i=1}^{N}(Q^*_E - Q_E)_i^2$ over $N$ stages of time where $(Q^*_E)_i$ is the known desired value of the effluent.

The material balance relations must be satisfied over $N$ stages of time:

$$\sum_{i=1}^{N}(Q_I)_i = \sum_{i=1}^{N}(Q_J)_i + A_2(H_1 - H_N) = \sum_{i=1}^{N}(Q_E)_i + A_1(h_1 - h_N) \quad (1)$$

where $A_1$ and $A_2$ are the cross-sectional areas of the lower and upper tanks. The $h_i$ and $H_i$ represent the heads at time $i$ in the lower and upper tanks.

The instantaneous material balance relations at the $i$th time stage are

$$(Q_I)_i = (Q_J)_i + A_2(H_i - H_{i-1}) \quad (2)$$

$$(Q_J)_i = (Q_E)_i + A_1(h_i - h_{i-1}) \quad (3)$$
The discharge flow rates for the two tanks are given by

\[(Q_j)_i = K_j \sqrt{H_i}\]  
\[(Q_E)_i = K_E \sqrt{h_i}\]  

(4)  
(5)

The system is constrained by

\[H_{\text{Min}} \leq H_i \leq H_{\text{Max}}\]  
\[h_{\text{Min}} \leq h_i \leq h_{\text{Max}}\]  
\[(Q_I)_{\text{Min}} \leq (Q_I)_i \leq (Q_I)_{\text{Max}}\]  
\[(Q^*_E)_i = C_i, \quad i = 1, 2, \ldots, N\]  

(6)  
(7)  
(8)  
(9)

We define

\[f_N(H, h) = \text{the value of the error squared between the desired effluent rate and the actual delivered rate, summed over the } N \text{ remaining stages of time, beginning in state } H \text{ for the upper tank, and in state } h \text{ for the lower tank, subject to the relations (1)-(9), and using an optimal policy}\]

Then

\[f_N(H, h) = \min_{(Q^*_I)_i} \sum_{i=1}^{N} (Q^*_E - Q_E)^2\]  

(10)

\[f_N(H, h) = \min_{(Q^*_I)_i} \left[ (Q^*_E - Q_E)^2 + \sum_{i=1}^{N-1} (Q^*_E - Q_E)^2 \right]\]  

(11)

and thus

\[f_N(H, h) = \min_{(Q^*_I)_i} \left[ (Q^*_E - Q_E)^2 + f_{N-1}(H', h') \right], \quad N \geq 2\]  

(12)

where \(H'\) and \(h'\) are the new values of \(H\) and \(h\) determined by (2)-(5),

\[f_1(H, h) = \min_{(Q^*_I)_i} [(Q^*_E - Q_E)^2]\]  

(13)

If the flow rates \(Q_J\) and \(Q_E\) were controlled by pumps, then Eqs. (4) and (5) no longer hold. They are replaced by some net positive suction head-flow relationship. In this case minimization is achieved by manipulating all three flows \((Q_I)_i, (Q_J)_i, (Q_E)_i\).

The functional equations have the form

\[f_N(H, h) = \min_{(Q^*_I)_i, (Q^*_J)_i, (Q^*_E)_i} \left[ (Q^*_E - Q_E)^2 + f_{N-1}(H', h') \right]\]  

(14)

\[f_1(H, h) = \min_{(Q^*_I)_i, (Q^*_J)_i, (Q^*_E)_i} [(Q^*_E - Q_E)^2]\]  

(15)
4. Multi-Tank Cascade System

The extension of the analysis above to a cascade system of $k$ tanks and $N$ stages of time may be written by analogy:

$$f_N(h_1, h_2, \ldots, h_k) = \min_{(h'_1, \ldots, h'_k)} \left[ (Q^*_E - Q_E)^2_N + f_{N-1}(h'_1, h'_2, \ldots, h'_k) \right]$$

where

$h_1, h_2, \ldots, h_k = \text{the levels in the tanks 1, 2, \ldots, } k \text{ at the beginning of time stage } N$

$h'_1, h'_2, \ldots, h'_k = \text{the new values of the levels in the tanks 1, 2, \ldots, } k \text{ at the beginning of time stage } (N - 1)$

$(Q_E)_i = \text{the effluent rate from the terminal tank at time stage } i$

The solution to this problem, due to its high dimensionality, in the state variables is by no means trivial. The general problem of high dimensionality in dynamic programming is discussed in Chapter 5.

5. Discussion of Cascade Tank Problem

The tank control problems discussed in Sections 2–4 are based on manipulating the inlet flow rate to adjust effluent flow rate for the terminal tank so it meets the performance criteria of $\sum_{i=1}^{N}(Q^*_E - Q_E)^2_i$. Depending on the problem and the objectives of control, a great many different performance indexes might have been used. Some of these are:

(a) $\min \sum_{i=1}^{N}(Q^*_E - Q_E)^2_i \text{ subject to } (Q^*_E - Q_E)^2_i \leq D_i$

This requires the error squared to be less than a constant at each stage. The constant may be different for each time stage. This handles the possibility that it may be more important to have closer control during part of the operating period than at others.

(b) $\min \sum_{i=1}^{N}|Q^*_E - Q_E|_i$

(c) $\min \sum_{i=1}^{N}(Q^*_E - Q_E)_i$

(d) $\min \sum_{i=1}^{N}\left(\frac{Q_E}{Q^*_E}\right)^2_i$
In some cases the situations of being above or below the desired point are not regarded with equal equanimity. It may turn out that being above the desired point, for example, is considered more serious than being below. In problems of this type, the relative bias can be weighted by using, say, a higher cost for positive error and lower cost for negative error. Instead of minimizing the error squared, \( \sum_{i=1}^{N} (Q_E^{*} - Q_E)^2 \), we would minimize the cost of the error squared. To be more specific, we minimize

\[
\sum_{i=1}^{N} v_i (Q_E^{*} - Q_E)^2
\]

where

\[
\begin{align*}
v_i &= a, & \text{if} & & (Q_E)_i > (Q_E^{*})_i \\
v_i &= 0, & \text{if} & & (Q_E)_i = (Q_E^{*})_i \\
v_i &= b, & \text{if} & & (Q_E)_i < (Q_E^{*})_i
\end{align*}
\]

Another variation of this theme is to make the cost of the deviation a function of the size of the deviation between the desired and actual flow. For example, we might desire to minimize

\[
\sum_{i=1}^{N} v_i (Q_E^{*} - Q_E)^2
\]

where the cost of the deviation is taken to be directly proportional to the size of the deviation. Here we have

\[
\begin{align*}
v_i &= a(Q_E^{*} - Q_E)_i, & \text{if} & & (Q_E)_i > (Q_E^{*})_i \\
v_i &= 0, & \text{if} & & (Q_E)_i = (Q_E^{*})_i \\
v_i &= b(Q_E^{*} - Q_E)_i, & \text{if} & & (Q_E)_i < (Q_E^{*})_i
\end{align*}
\]

6. Terminal Control of a Tubular Reactor

a. Reactor Equations

We will develop the equations for the thermal cracking of vapor phase reaction in a tubular reactor. The nomenclature is defined below.

The reaction is described by compound \( A \) cracking to compounds \( B \) and \( D \):

\[
A \xrightarrow{\Delta} B + D
\]

(1)
The volumetric flow rate per unit time is
\[ \frac{znRT}{P} = A_c \frac{dl}{dt} \] (2)

The average residence time is
\[ dt = \frac{A_c dl}{znRT/P} = \frac{A_c dl}{z(W + X)RT/P} \] (3)

The rate of reaction is expressed as
\[ \frac{dX}{dt} = k_1 C_A - k_2 C_B C_D \] (4)

where
\[ C_A = \left( \frac{W - X}{W + X} \right) \frac{P}{RTz} \] (5)
\[ C_B = \left( \frac{X}{W + X} \right) \frac{P}{RTz} \] (6)
\[ C_D = \left( \frac{X}{W + X} \right) \frac{P}{RTz} \] (7)
\[ k_1 = k_0 e^{-E/RT} \] (8)
\[ \frac{k_1}{k_2} = K_c \] (9)

Combining (3)–(7) we obtain
\[ dt = \frac{dX}{k_1 \left( \frac{W - X}{W + X} \right) \frac{P}{RTz} - k_2 \left( \frac{X}{W + X} \right)^2 \left( \frac{P}{RTz} \right)^2} \] (10)

or
\[ \frac{(W + X) dX}{k_1 \left( \frac{W - X}{W + X} \right) \left( \frac{P}{RTz} \right)^2 - k_2 \left( \frac{X}{W + X} \right)^2 \left( \frac{P}{RTz} \right)^3} = A_c \frac{dl}{dl} \] (11)

Integrating over the reactor tube length, we have
\[ \int_{X(0)}^{X(L)} \frac{(W + X) dX}{k_1 \left( \frac{W - X}{W + X} \right) \left( \frac{P}{RTz} \right)^2 - k_2 \left( \frac{X}{W + X} \right)^2 \left( \frac{P}{RTz} \right)^3} = A_c \int_0^L dl = A_c L \] (12)
The differential pressure drop is evaluated by

\[ \frac{dP}{dl} = \frac{2e(W + X)^2 M^2}{\rho d \delta c} \]  

(13)

A heat balance yields

\[ \frac{dQ}{dl} = (W + X) M c_p \frac{dT}{dl} + H \frac{dX}{dl} \]  

(14)

The vapor and kinetic properties \( c_p, H, k_1, k_2, M, z, \) and \( \rho \) are known functions of temperature.

The nomenclature is:

- \( A \) = Compound \( A \)
- \( A_e \) = Cross-sectional area of tube
- \( B \) = Compound \( B \)
- \( C \) = Conversion, fraction of compound \( A \) that reacts \( = X/W \)
- \( C_A \) = Concentration of compound \( A \)
- \( C_B \) = Concentration of compound \( B \)
- \( C_D \) = Concentration of compound \( D \)
- \( c_p \) = Specific heat of gas mixture
- \( D \) = Compound \( D \)
- \( d \) = Tube diameter
- \( e \) = Friction factor
- \( E \) = Energy of activation
- \( g \) = Gravitational constant
- \( H \) = Heat of reaction
- \( k_o \) = Arrhenius constant
- \( k_1 \) = Forward reaction velocity constant
- \( k_2 \) = Reverse reaction velocity constant
- \( K_e \) = Equilibrium constant \( = k_1/k_2 \)
- \( M \) = Molecular weight of gas mixture
- \( n \) = Total rate of gas flowing in reactor, mol/time \( = W + X \)
- \( P \) = Pressure in reactor
- \( Q \) = Heat transferred
- \( l \) = Tube length
- \( L_e \) = Total tube length
- \( t \) = Time
- \( T \) = Temperature of mixed gases
- \( W \) = Flow rate of \( A \) entering reactor, mol/time
- \( X \) = Product rate of compound \( B \), mol/time, or product rate of compound \( D \), or mol/time of reactant \( A \) that reacts
- \( z \) = Compressibility factor of mixed gases
- \( \rho \) = Density of mixed gases

b. Terminal Control Equations

We desire to control this reactor by manipulating the heat input as a function of tube length so that the squared difference between the conversion at the reactor outlet and a desired conversion at the outlet is minimized.

That is to say, we desire to minimize \( J(Q) \):

\[ J(Q) = (C(L) - C_d)^2 \]  

(15)

where

\[ C(L) = \text{Conversion at the reactor outlet} \]
\[ C_d = \text{Desired conversion at the reactor outlet} \]

\[ C = X/W \]  

(16)
Since $W$ is a known constant, we may express the criterion function
alternately as

$$J(Q) = (X(L) - X_d)^2$$

(17)

If we require that the maximum amount of compound $A$ that can
react is to be equal to or less than $\beta$, we may express this limit as

$$\int_{X(0)}^{X(L)} dX \leq \beta, \quad \beta \leq W$$

(18)

To take into account (18), which represents an isoperimetric constraint,
we redefine the $J(Q)$ function as

$$J(Q) = (X(L) - X_d)^2 + \lambda \int_{X(0)}^{X(L)} dX$$

(19)

where

$\lambda$ = Lagrangian multiplier

$X(0)$ = Value of $X$ at the inlet to the reactor

$X(L)$ = Value of $X$ at the outlet of the reactor

$X_d$ = Desired value of $X$ at the outlet of the reactor

In this formulation, we note that $Q$ does not appear explicitly. The
effect of $Q$ is manifested by Eqs. (11)–(14) in terms of $P$, $T$, $\rho$, $M$, $X$,
and the fluid properties.

We now divide the reactor tube length into $N$ equal elements of
length $\Delta l$ so that $N\Delta l = L$. Corresponding to each element of the
tube there is a differential quantity of reactant $A$ that is converted,
namely, $\Delta X_k$. The total quantity of $A$ that reacts over the tube is
$\Sigma_{k=1}^{N} \Delta X_k$. In this expression, the $\Delta X_k$ are not taken to be of equal size.
The size of the $\Delta X_k$ is determined from the finite difference equations
given below. The elemental change $\Delta X_k$ takes place in the elemental
length $(\Delta l)_k$. Since the elemental lengths $(\Delta l)_k$ are of equal size, the
subscripts will be omitted.

The finite difference expressions corresponding to (11), (12), (13),
(14), (18), and (19) are given in Eqs. (20), (21), (22), (23), (24), and (25),
respectively:

$$k_1 \left( \frac{W - X_k}{W + X_k} \right)^2 \left( \frac{P_{k-1}}{RT_{k-1}z_{k-1}} \right)^2 - k_2 \left( \frac{X_k}{W + X_k} \right)^2 \left( \frac{P_{k-1}}{RT_{k-1}z_{k-1}} \right)^3 = A_c \Delta l$$

(20)

$$\frac{\Delta P_k}{\Delta l} = \left[ \frac{2\varepsilon_{k-1}(W + X_k)^2 M_{k-1}^2}{\rho_{k-1} A_c^2 \Delta l} \right]$$

(22)
Within an elemental $\Delta l$, the values of $\Delta X_k$ obtained from Eqs. (20) and (23) must be reconciled. Rearranging (20) and (23) we find

$$\Delta X_k = \left( \frac{W - X_k}{W + X_k} \right)^2 \left( \frac{P_k}{RT_{k-1} z_{k-1}} \right) - k_2 \left( \frac{X_k}{W + X_k} \right)^2 \left( \frac{P_k}{RT_{k-1} z_{k-1}} \right)^3$$

Equating (26) and (27), we have

$$\frac{\Delta T_k}{\Delta l} = \frac{1}{H_{k-1}} \left[ \frac{\Delta Q_k}{\Delta l} - (W + X_k) M_{k-1} (c_p)_{k-1} \frac{\Delta T_k}{\Delta l} \right]$$

We shall use (26) and (27) to determine the value of $\Delta X_k$ corresponding to $\Delta Q_k$.

The fluid properties are to be evaluated at the downstream side of each $\Delta l$, which accounts for the way Eqs. (20)–(24) and (26)–(28) have been written.

In the above equations, we define

$$X_{k-1} = X_k + \Delta X_k$$

$$T_{k-1} = T_k + \Delta T_k$$

$$P_{k-1} = P_k - \Delta P_k$$

The convention adapted for counting the stages is shown in Fig. 3. The following information is known:

(a) $P_o$, $T_o$, the outlet reactor pressure and temperature

(b) $W$, the flow rate of compound $A$ entering the reactor

(c) $X_N = b$, the initial value of $X_N$

(d) All physical and chemical properties of the mixed gas will be evaluated at the downstream side of each $\Delta l$
The following information is assumed and must be checked:
$M_o, z_o, (c_p)_o$, etc. (the properties of the mixed gases at the outlet, which depend upon $P_o, T_o$, and the composition of the mixed gases).

We now define

$$f_N(b) = \min_{\Delta Q_k} J_N(\Delta Q_k) = \text{the value of the squared error at the outlet of the reactor between the actual mols of compound } A \text{ that have reacted and the desired reacted mols of compound } A \text{ beginning in the state } X_N = b \text{ and following an optimal policy}$$

We assume that $\Delta Q_k$ is bounded

$$\Delta Q_k \leq \Delta Q_{\text{Max}}$$

From the definition in Eq. (32), it follows that

$$f_N(b) = \min_{\Delta Q_k} \left[ (X_0 - X_d)^2 + \lambda \sum_{k=1}^N \Delta X_k \right]$$

$$f_N(b) = \min_{\Delta Q_N} [\lambda \Delta X_N + f_{N-1}(b + \Delta X_N)]$$

$$f_1(b) = \min_{\Delta Q_1} [\lambda \Delta X_1 + (X_0 - X_d)^2]$$

This problem is slightly different from others presented in this book since the manipulated variable, the $\Delta Q_k$ does not appear explicitly in the functional equations.

The computational procedure will be described for the one-stage and two-stage processes. From this the $N$-stage calculation procedure clearly follows. First of all the Lagrangian multiplier is determined by the usual trial and error process described in Chapter 5, Section 19. When the constraint [Eq. (24)] is satisfied, the proper $\lambda$ is found. For the one-stage process we know $P_o, T_o, W$, and $b$. In order to calculate at the reactor outlet the physical properties of the mixed gases which depend upon $P_o, T_o$, and the composition, we assume a terminal composition. With values for $\lambda, P_o, T_o, W, b, e_o, \rho_o, H_o, M_o, (c_p)_o$, and $z_o$ known or assumed, we choose arbitrarily a $\Delta Q_1$, and solve Eqs. (28) and (30) for $\Delta T_1$ and $T_1$. Using $\Delta T_1$ in (27), we find $\Delta X_1$. From (29) we find $X_o$ where

$$X_o = b + \Delta X_1$$
We then evaluate the bracketed term in (36). This process is repeated for various values of $\Delta Q_1$ until a minimum value for the right-hand side of (36) is achieved. Once this is found, $\Delta P_1$ and $P_1$ are evaluated by (22) and (31). From the $P_1$ and $T_1$ the various fluid properties are evaluated at stage 1. At this point it is necessary to calculate the properties of the mixed gases at the reactor outlet to check the assumed values or rather the effects of the assumed values. Although the physical properties of the mixed gases are complicated functions of pressure, temperature, and composition, it is not necessary that an exact match be made. All that is required is that the assumed and calculated set of physical properties for the mixed gases at the reactor outlet generate essentially the same values of $P_1$, $T_1$, and $\Delta X_1$. If a mismatch occurs, the process must be repeated for another assumed set of physical properties of the mixed gases at the reactor outlet. The one-stage process is carried out for a number of discrete values of $b$; namely, $b_0$, $b_1$, $b_2$, ..., where $b_0 < b_1 < b_2$.

For the two-stage process, corresponding to each discrete value of $b$ used in the one-stage process, the values of $P_1$, $T_1$, $W$, $e_1$, $\rho_1$, $H_1$, $M_1$, etc., are known. For each initial value $X_2 = b$ beginning the two-stage process, the $\Delta Q_2$ must be chosen to minimize the right-hand side of Eq. (35). Since $f_1(b + \Delta X_1)$ in Eq. (35) has already been evaluated by the one-stage process, only the first term on the right-hand side of Eq. (35) need be calculated. This is done by arbitrarily choosing a value of $\Delta Q_2$ and solving Eqs. (28) and (30) for $\Delta T_2$ and $T_2$ since $P_1$, $T_1$, $W$, $e_1$, $\rho_1$, $H_1$, $M_1$, and $X_2 = b$ are known. From Eq. (27), $\Delta X_2$ is found and then from Eq. (29), $X_1$ is obtained. The calculation procedure is repeated for various values of $\Delta Q_2$ until a minimum is found for Eq. (35). In short, the search for the minimizing value of $\Delta Q_2$ is identical to that for the search for $\Delta Q_1$ in the one-stage process. Once again we have a problem of determining downstream physical properties—in this case, which values of $P_1$, $T_1$, $e_1$, $\rho_1$, $H_1$, $M_1$, etc., to use for stage two of the two-stage calculation. For each discrete value of $b$ used in the one-stage process a corresponding set of $P_1$, $T_1$, $e_1$, $\rho_1$, $H_1$, $M_1$, etc., is developed for the minimizing $\Delta Q_1$. To start the calculation for stage two for each value of $X_2 = b$, a set of $P_1$, $T_1$, $e_1$, $\rho_1$, $H_1$, $M_1$, etc., must be chosen. Since it is not known beforehand whether the minimizing $\Delta Q_2$ will drive the process from, say $X_2 = b_3$ to $X_1 = b_4$ or $b_5$ or $b_6$, etc., or to some intermediate $b$ value, the proper set of $P_1$, $T_1$, $e_1$, $\rho_1$, $H_1$, $M_1$, etc., is not known for sure beforehand. This means that after the minimizing value of $\Delta Q_2$ is found a check must be made for the assumed set of $P_1$, $T_1$, $e_1$, $\rho_1$, etc., to be sure that the value of $X_1$ and its corresponding physical properties are reasonably close to the
7. Averaging Control for Tubular Reactor

For the tubular reactor described in Section 6, we desire to maximize the conversion of reactant $A$ by manipulating the heat flux as a function of reactor tube length. We wish then to maximize $J(Q)$,

$$J(Q) = \int_{C(0)}^{C(L)} dC$$  \hspace{1cm} (1)

where

$$C(0) = \text{Conversion at the inlet to the reactor}$$

$$C(L) = \text{Conversion at the outlet of the reactor}$$

Since $C = X/W$ and $W$ is a known constant, we write (1) as

$$J(Q) = \frac{1}{W} \int_{X(0)}^{X(L)} dX = \frac{1}{W} [X(L) - X(0)]$$  \hspace{1cm} (2)

or

$$J(Q) = \frac{1}{W} \int_{0}^{L} \frac{dX}{dL} dL$$  \hspace{1cm} (3)

Using discrete approximations, we replace (3) by

$$J_N(\Delta Q_k) = \frac{1}{W} \sum_{k=1}^{N} \frac{\Delta X_k}{\Delta l} \Delta l, \quad k = 1, 2, ..., N$$  \hspace{1cm} (4)

Recognizing that the maximum amount of compound $A$ that can react is equal to or less than $\beta$, we express this limit as

$$\int_{X(0)}^{X(L)} dX \leq \beta, \quad \beta \leq W$$  \hspace{1cm} (5)

or, in discrete form, as

$$\sum_{k=1}^{N} \Delta X_k \leq \beta, \quad \beta \leq W$$  \hspace{1cm} (6)
Taking into account the isoperimetric constraint, Eq. (20), (6), and using discrete approximations, we redefine our objective to maximize $J_N(\Delta Q_k)$:

$$J_N(\Delta Q_k) = \sum_{k=1}^{N} \left\{ \frac{1}{W} \cdot \frac{\Delta X_k}{\Delta l} \Delta l + \lambda \Delta X_k \right\}$$

where $\lambda$ is a Lagrangian multiplier.

We assume that $\Delta Q_k \leq \Delta Q_{\text{Max}}$ (8)

The initial inlet condition is given by $X_N = b$ (9)

We define $f_N(b) = \text{Max}_{\Delta Q_k} J_N(\Delta Q_k) = \text{the conversion over the reactor of length } N \Delta l \text{ beginning with } b \text{ moles/time of compound } A \text{ that have reacted,}$

and using an optimal policy

The functional equations are

$$f_N(b) = \text{Max}_{\Delta Q_k} \left[ \sum_{k=1}^{N} \frac{1}{W} \cdot \frac{\Delta X_k}{\Delta l} \Delta l + \lambda \Delta X_k \right]$$

(11)

$$f_N(b) = \text{Max}_{\Delta Q_k} \left[ \frac{\Delta X_N}{W} + \lambda \Delta X_N + f_{N-1}(b + \Delta X_N) \right]$$

(12)

$$f_1(b) = \text{Max}_{\Delta Q_1} \left[ \frac{\Delta X_1}{W} + \lambda \Delta X_1 \right]$$

(13)

The counting convention for stages as well as the known and assumed information in Section 6 apply here.

The computational procedure is carried out identically to that described for the terminal control problem in Section 6. The only difference is the maximization of the functional equations rather than the minimization.

It is interesting to compare the treatment of this discrete dynamic programming problem with a somewhat similar problem in Chapter 4, Section 22. In Chapter 4, Section 22 a continuous process was used, employing the notation $f(x_o, y_o, T)$ where $x_o, y_o$ were the initial concentrations of the reactants and $T$ was the time duration of the process. If a discrete process had been used, the function would have been written as $f_M(x_o, y_o)$ where $M$ refers to the number of time stages remaining. This is identically the same notation used in this section. There are two differences between Chapter 4, Section 22 and this section. The first difference is that here the stages refer to stages of reactor tube length rather than stages of time. The second difference is that in this chemical reaction only one state variable is required.
8. Method of Kalman et al.

In a series of papers by Kalman, Lapidus, E. Shapiro, and their co-workers, the control of chemical processes by dynamic programming is discussed. The problem solved by these investigators is the control of a chemical process from an initial steady state or equilibrium point to a final equilibrium point. The original equations are linearized around the equilibrium point. The control criterion, or performance index, employed is a quadratic function of the difference between the desired state and the actual state. The error function (performance index) is minimized over \( N \) stages of time by the methodology of dynamic programming.

We will first discuss and expand the paper by Lapidus and co-workers [6] and then cover somewhat the same material by a slightly different approach of Kalman. To round out the theoretical treatment, an example of this technique is given.

9. Linearization around Equilibrium

We consider a set of nonlinear differential equations that describe a process as

\[
\frac{dy_i}{dt} = f_i(y_1, y_2, ..., y_n, t) + g_i(F_1, F_2, ..., F_m, y_{01}, y_{02}, ..., y_{0n}, t)
\]

where

- \( y_i \) = the process variables, \( i = 1, 2, ..., n \)
- \( y_{0i} \) = the initial values of \( y_i, i = 1, 2, ..., n \)
- \( F_i \) = the controllable or manipulated variables, \( i = 1, 2, ..., m \)

The right-hand side of the equation consists of two parts; the \( f_i \) term, which is the free or unforced term and a forcing term \( g_i \). The forcing term is altered by manipulating the controllable variables to meet some criterion of performance.

For small perturbations around an equilibrium point the generalized nonlinear differential equations (1) may be linearized by the device

\[
y - y_i^* = x_i
\]

\[
F - F_i^* = m_i
\]

where \( y_i^* \), \( F_i^* \) represent equilibrium values and \( x_i \) and \( m_i \) represent the deviations from equilibrium.
The set of equations (1) is conveniently and compactly written in vector notation
\[
\frac{dy}{dt} = f(y, t) + g(F, y_0, t) \tag{4}
\]

Using a Taylor series expansion up to linear terms, we write
\[
f(y, t) = f(y^* + x, t) = f(y^*, t) + x \left( \frac{\partial f(y^*, t)}{\partial y^*} \right)_{y=y^*} \tag{5}
\]
\[
g(F, y_0, t) = g(F^* + m, y_0^* + x, t) \tag{6}
\]
\[
g(F, y_0, t) = g(F^*, y_0^*, t) + m \left\{ \frac{\partial g(F^*, y_0^*, t)}{\partial F^*} \right\}_{F=F^*, y=y_0^*} + x \left\{ \frac{\partial g(F^*, y_0^*, t)}{\partial y_0^*} \right\}_{y=y_0^*} \tag{7}
\]

At equilibrium
\[
\frac{dy^*}{dt} = f(y^*, t) + g(F^*, y_0^*, t) = 0 \tag{8}
\]
\[
\frac{dy}{dt} - \frac{dy^*}{dt} = \frac{dx}{dt} \tag{9}
\]

Substituting Eqs. (5), (7), and (9) into Eq. (4), we see that
\[
\frac{dx}{dt} + \frac{dy^*}{dt} = f(y^*, t) + x \frac{\partial f(y^*, t)}{\partial y^*} + g(F^*, y_0^*, t) + m \frac{\partial g(F^*, y_0^*, t)}{\partial F^*} + x \frac{\partial g(F^*, y_0^*, t)}{\partial y_0^*} \tag{10}
\]

Subtracting Eq. (8) from (10) gives
\[
\frac{dx}{dt} = x \frac{\partial f(y^*, t)}{\partial y^*} + m \frac{\partial g(F^*, y_0^*, t)}{\partial F^*} + x \frac{\partial g(F^*, y_0^*, t)}{\partial y_0^*} \tag{11}
\]

If we assume that \(y_0^*\) the vector of the initial values of the process variables, is a constant, the last term in Eq. (11) drops out. Equation (11) reduces to
\[
\frac{dx}{dt} = Ax + Dm \tag{12}
\]

The elements of \(A\) are
\[
\left( \frac{\partial f_i}{\partial y_j} \right)_{y=y^*}; \quad i = 1, 2, ..., n \quad j = 1, 2, ..., n \tag{13a}
\]
and the elements of $D$ are

$$
\left( \frac{\partial g_i}{\partial F_j} \right)_{F=F^*}, \quad i = 1, 2, \ldots, n \quad j = 1, 2, \ldots, m
$$

(13b)

The terms in Eq. (11) are described as follows:

- $x$ = $n$-dimensional vector
- $A$ = $n \times n$ matrix
- $D$ = $n \times m$ matrix
- $m$ = $m$-dimensional vector

To review what we have done up to this point, we started with a set of nonlinear differential equations describing a chemical process. The set of equations were linearized by assuming small deviations from the equilibrium point. The original perturbation equations (2) and (3), the Taylor series expansion equations (5)–(7) and the conditions for equilibrium equations (8) and (9) were all used to convert the set in Eq. (1) into the linearized set, with constant coefficients, Eq. (12). The linearized set is linear in terms of the deviations from equilibrium. The vectors $x$ and $m$ are functions of time.

The differential matrix equation, Eq. (12), integrates to

$$
x(t) = e^{At}x(0) + \int_0^t e^{A(t-\lambda)} Dm \, d\lambda
$$

(14)

where

$$
x(0) = x(t) \quad \text{at} \quad t = 0
$$

(15)

The term $e^{At}x(0)$ is the solution of

$$
\frac{dx}{dt} = Ax, \quad x(0) = I
$$

(16)

The matrix exponential term $e^{At}$ may be evaluated in the same manner as a scalar exponential term, namely, by a series expansion:

$$
e^{At} = I + \frac{At}{1!} + \frac{(At)^2}{2!} + \ldots + \frac{(At)^n}{n!} + \ldots
$$

(17)

If it is assumed that the inputs or manipulated variables can be changed only at discrete time intervals $0, \tau, 2\tau, \ldots, k\tau$ where $\tau$ is the sampling period, and are held constant within each interval, then the solution of Eq. (12) may be written as

$$
x(k + 1) = \Phi x(k) + \Delta m(k)
$$

(18)
where

\[ \Phi = e^{At} \]  
(19)

\[ \Delta = \int_0^T e^{A(t-\lambda)} D d\lambda \]  
(20)

The matrices \( \Phi \) and \( \Delta \) are not functions of time since \( A \) and \( D \) have constant elements. The term \( x(k) \) refers to the value of \( x(t) \) at the \( k \)th sampling instant. The term \( \Phi \) is sometimes called the transition matrix. For a system with no forcing function, the transition matrix operating on \( x(k) \) produces the next term; namely, \( x(k + 1) \).

The significance of Eqs. (19) and (20) is that the \( \Phi \) and \( \Delta \) can be evaluated numerically once and for all. The \( A \) and \( D \) matrices are known from Eq. (12). The quantity \( \Phi \) can be evaluated from Eqs. (17) and (19) where \( \tau \) replaces \( t \). Once \( \Phi \) is known, Eq. (20) may be solved. With \( \Phi \) and \( \Delta \) evaluated numerically, Eq. (18) may be solved for \( x(k + 1) \) given \( x(k) \) and the value of the manipulated vector \( m(k) \).

\[ J(m(k - 1)) = \sum_{i=1}^N [x^d - x(k)]^T Q[x^d - x(k)] \]  
(1)

where \( x^d \) is the desired state vector, \( x(k) \) is the state vector at sampling instant \( k \), and \( (x^d - x(k))^T \) is the transpose of the error. The minimization is to be executed by the manipulation of the control vector \( m(k - 1) \). The matrix \( Q \) is a symmetric positive semidefinite matrix. A positive semidefinite matrix \( Q \) has the property that the quadratic form \( z^T Q z \) is non-negative for all real values of \( z_i \), the elements of \( z \). By suitable transformations, a positive semidefinite matrix may be reduced to a diagonal matrix. Algebraically, this means that the quadratic form \( z^T Q z \) which contains cross product as well as squared terms may be transformed into an expression with only squared terms. Geometrically,
this is equivalent to rotation and translation of the quadratic surface so that the principle axes of the quadratic are parallel to the coordinate axes. The significance of the use of $Q$, a positive semidefinite matrix, is that the quadratic form Eq. (1) is a number which is greater than or equal to zero.

The utilization of a quadratic form or any other performance index is a subjective matter which must be decided by the investigator. In this case, the matrix $Q$ is used to weight the relative importance of the cross product and squared terms.

For example, if

$$Q = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{pmatrix}$$

then

$$[x_1^d - x_1 \quad x_2^d - x_2 \quad x_3^d - x_3] \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{pmatrix} \begin{pmatrix} x_1^d - x_1 \\ x_2^d - x_2 \\ x_3^d - x_3 \end{pmatrix} = (x_1^d - x_1)^2 + 2(x_2^d - x_2)^2 + 3(x_3^d - x_3)^2$$

In this example, it is seen that the deviation of $x_3$ from its desired value costs more than other deviations since its coefficient 3 is the largest. If $Q = I$, then the cost of deviation for $x_1$, $x_2$, $x_3$ is the same. The coefficients in the expanded quadratic form are numbers and $[x^d - x]^T Q [x^d - x] \geq 0$ since $Q$ is positive semidefinite.

We will take advantage of the symmetry property of $Q$ when the quadratic form is differentiated.

11. Control Equations

In this process, we count time forward so that $k = 0$ refers to the beginning of the process and $k = N$ refers to the end of the process.

We may write the performance index $J$ as

$$J(m(k - 1)) = \sum_{k=1}^{N} [x^d - x(k)]^T Q [x^d - x(k)]$$

(1)
We define a sequence of functions

\[ F_{N-1}(x) = \min_{m(N-1)} J(m(N-1)) = \min_{m(N-1)} [x^d - x(N)]^T Q [x^d - x(N)] \]  

(2)

= the value of the quadratic form over the last stage of the process, \((N - 1)\) to \(N\) beginning in state \(x(N - 1)\), and using an optimal policy

\[ F_{N-2}(x) = \min_{m(N-2)} J(m(N-2)) = \min_{m(N-2)} \sum_{k=N-1}^{N} [x^d - x(k)]^T Q [x^d - x(k)] \]  

(3)

= the value of the quadratic form over the last two stages \((N - 2)\) to \(N\), starting in state \(x(N - 2)\), and using an optimal policy

A general expression for the stage \((N - j)\) measured from the end of the process is

\[ F_{N-j}(x) = \min_{m(N-j)} \left\{ \sum_{k=N-j+1}^{N} [x^d - x(k)]^T Q [x^d - x(k)] \right\} \]  

(4)

\[ F_{N-j}(x) = \min_{m(N-j)} ([x^d - x(N - j + 1)]^T Q [x^d - x(N - j + 1)] + F_{N-j+1}(x)) \]  

(5)

In the expression \(F_{N-j}(x)\), the state vector \(x\) refers to \(x(N - j)\) and similarly in \(F_{N-j+1}(x)\), the state vector \(x\) refers to \(x(N - j + 1)\).

In these definitions it is important to remember that the manipulated vector \(m(N - j)\) affects the state of the system at all succeeding time periods, namely, \(N - j + 1, N - j + 2, \ldots, N\). It does not affect the state of the system at previous times, namely, \(N - j - 1, N - j - 2, \ldots, 0\) (see Fig. 4).

![Fig. 4.](image)

For example, referring to Eq. (2), to minimize the performance index at stage \(N\), the appropriate control action beginning at stage \((N - 1)\) is \(m(N - 1)\) acting on \(x(N - 1)\). The control action \(m(N - 1)\) has no effect whatsoever on \(x(N - 2), x(N - 3), \ldots, x(0)\). Referring to Eq. (3) the control action \(m(N - 2)\) affects \(x(N - 1)\) and \(x(N)\) but does not affect \(x(N - 2), x(N - 3), \ldots, x(0)\).

The minimization over the last stage \((N - 1)\) to \(N\) might be made
11. CONTROL EQUATIONS

by a search of values for $m(N - 1)$. This is not necessary in this case since the quadratic expression can be differentiated and the derivative set equal to zero.

For the last stage $(N - 1)$ to $N$, we carry out the minimization in Eq. (2). Setting

$$\frac{\partial}{\partial m(N - 1)} [x^d - x(N)]^T Q [x^d - x(N)] = 0$$  \hspace{1cm} (6)

we have

$$\frac{\partial f(m(N - 1))}{\partial m(N - 1)} = -2 \frac{\partial x^T(N)}{\partial m(N - 1)} Q [x^d - x(N)]$$  \hspace{1cm} (7)

Using Eq. (18), Section 9, we can show

$$\frac{\partial x^T(N)}{\partial m(N - 1)} = A^T$$  \hspace{1cm} (8)

Substituting (8) into (9) we obtain

$$A^T Q [x^d - x(N)] = 0$$  \hspace{1cm} (9)

The symmetry property of the matrix $Q$ accounts for the simple derivative equation (7).

Rewriting Eq. (18) of Section 9 with the argument $N = k + 1$, we have

$$x(N) = \Phi x(N - 1) + \Delta m(N - 1)$$  \hspace{1cm} (10)

On substituting Eq. (10) into (9) and rearranging, we find that

$$m(N - 1) = C^{(N-1)} x(N - 1) + H^{(N-1)} x^d$$  \hspace{1cm} (11)

where

$$C^{(N-1)} = -(A^T Q A)^{-1} A^T Q \Phi$$  \hspace{1cm} (12)
$$H^{(N-1)} = (A^T Q A)^{-1} A^T Q$$  \hspace{1cm} (13)

Substituting Eq. (11) into (10) and collecting terms, we have

$$x(N) = [\Phi + \Delta C^{(N-1)}] x(N - 1) + \Delta H^{(N-1)} x^d$$  \hspace{1cm} (14)

Let

$$\psi^{(N-1)} = \Phi + \Delta C^{(N-1)}$$  \hspace{1cm} (15)
$$x(N) = \psi^{(N-1)} x(N - 1) + \Delta H^{(N-1)} x^d$$  \hspace{1cm} (16)
Equation (11) represents the value of $m(N-1)$ which executes the minimization in Eq. (2). This value of $m(N-1)$ acting on the state variable $x(N-1)$ in Eq. (2) generates the $x_N$ described by Eq. (14) or (16).

For the minimization over the last two time stages $(N-2)$ and $(N-1)$, we write Eq. (3) as

$$F_{N-2}(x) = \min_{m(N-2)} \left\{ [(x^d - x(N-1))^T Q [x^d - x(N-1)] + [x^d - x(N)]^T Q [x^d - x(N)]] \right\}$$

(17)

Taking the derivative of the right-hand side of Eq. (17), we have

$$\frac{\partial x^T(N-1)}{\partial m(N-2)} Q [x^d - x(N-1)] + \frac{\partial x^T(N)}{\partial m(N-2)} Q [x^d - x(N)] = 0$$

(18)

Substituting Eq. (16) into (18) and collecting terms, we find

$$\frac{\partial x^T(N-1)}{\partial m(N-2)} [-Q - \psi^{(N-1)T} \psi^{(N-1)}] x(N-1) + \frac{\partial x^T(N)}{\partial m(N-2)} [Q + \psi^{(N-1)T} Q - \psi^{(N-1)T} \Delta H^{(N-1)}] x^d = 0$$

(19)

Let

$$P^{(1)} = \psi^{(N-1)T} \psi^{(N-1)} + Q$$

$$K^{(N-1)} = \psi^{(N-1)T} Q - \psi^{(N-1)T} \Delta H^{(N-1)} + Q$$

(20)

(21)

Replacing the argument $N$ by $(N-1)$, we have from Eq. (10)

$$x(N-1) = \Phi x(N-2) + \Delta m(N-2)$$

(22)

Since $\partial x^T(N-1)/\partial m(N-2) = \Delta^T$, Eq. (19), using Eqs. (20)–(22), becomes

$$-\Delta^T P^{(1)} [\Phi x(N-2) + \Delta m(N-2)] + \Delta^T K^{(N-1)} x^d = 0$$

(23)

$$m(N-2) = (\Delta^T P^{(1)} \Delta)^{-1} (\Delta P^{(1)} \Phi) x(N-2) + (\Delta^T P^{(1)} \Delta)^{-1} \Delta^T K^{(N-1)} x^d$$

(24)

Let

$$C^{(N-2)} = -(\Delta^T P^{(1)} \Delta)^{-1} \Delta^T P^{(1)} \Phi$$

(25)

and

$$H^{(N-2)} = (\Delta^T P^{(1)} \Delta)^{-1} \Delta^T K^{(N-1)}$$

(26)

then

$$m(N-2) = C^{(N-2)} x(N-2) + H^{(N-2)} x^d$$

(27)
The substitution of (27) into (22) gives for $x(N - 1)$

$$x(N - 1) = \Phi x(N - 2) + \Delta [C^{(N-2)}] x(N - 2) + H^{(N-2)} x^d$$

(28)

$$x(N - 1) = [\Phi + \Delta C^{(N-2)}] x(N - 2) + \Delta H^{(N-2)} x^d$$

(29)

$$x(N - 1) = \psi^{(N-2)} x(N - 2) + \Delta H^{(N-2)} x^d$$

(30)

where

$$\psi^{(N-2)} = \Phi + \Delta C^{(N-2)}$$

(31)

Equation (27) gives the value of $m(N - 2)$ which determines $F_{N-2}(x)$ by minimizing the right-hand side of (3). Equation (29) yields the value of $x(N - 1)$ generated from $x(N - 2)$ by the $m(N - 2)$ in (27).

Continuing once again, we write for

$$F_{N-3}(x) = \min_{m(N-3)} \{[x^d - x(N - 2)]^T Q[x^d - x(N - 2)] + F_{N-2}(x)\}$$

(32)

The substitution of (17) for $F_{N-2}(x)$ into (32) provides

$$F_{N-3}(x) = \min_{m(N-3)} \{[x^d - x(N - 2)]^T Q[x^d - x(N - 2)]$$

$$+ [x^d - x(N - 1)]^T Q[x^d - x(N - 1)]$$

$$+ [x^d - x(N)]^T Q[x^d - x(N)]\}$$

(33)

The derivative of the right-hand side yields

$$\frac{\partial x^T(N)}{\partial m(N - 3)} Q[x^d - x(N - 2)] + \frac{\partial x^T(N - 1)}{\partial m(N - 3)} Q[x^d - x(N - 1)]$$

$$+ \frac{\partial x^T(N - 2)}{\partial m(N - 2)} Q[x^d - x(N - 2)] = 0$$

(34)

Replacing $x(N)$ by (16) and $x(N - 1)$ by (30) we obtain from (34) an equation in $x(N - 2)$

$$\frac{\partial x^T(N - 2)}{\partial m(N - 3)} \{\psi^{(N-2)} T \psi^{(N-1)} T$$

$$\times Q [x^d - \psi^{(N-1)}(\psi^{(N-2)} x(N - 2) + \Delta H^{(N-2)} x^d) - \Delta H^{(N-1)} x^d]$$

$$+ \frac{\partial x^T(N - 2)}{\partial m(N - 3)} \{\psi^{(N-2)} T Q[x^d - \psi^{(N-2)} x(N - 2) - \Delta H^{(N-2)} x^d]\}$$

$$+ \frac{\partial x^T(N - 2)}{\partial m(N - 3)} Q[x^d - x(N - 2)] = 0$$

(35)
Collecting terms in $x(N - 2)$ and $x^d$, we have

$$-\frac{\partial x^T(N - 2)}{\partial m(N - 3)} \{\psi^{(N-2)}_T \psi^{(N-1)} T Q \psi^{(N-1)} + \psi^{(N-2)}_T Q \psi^{(N-2)} + Q\} x(N - 2)$$

$$+ \frac{\partial x^T(N - 2)}{\partial m(N - 3)} \{\psi^{(N-2)}_T \psi^{(N-1)} T Q [I - \psi^{(N-1)} \Delta H^{(N-2)} - \Delta H^{(N-1)}]$$

$$+ \psi^{(N-2)}_T Q (I - \Delta H^{(N-2)}) + Q\} x^d = 0$$

Equation (36) is now written

Let

$$P^{(2)} = \psi^{(N-2)}_T \{\psi^{(N-2)} T Q \psi^{(N-1)} + Q\} \psi^{(N-2)} + Q$$

$$P^{(2)} = P^{(1)} + Q$$

Let

$$K^{(N-2)} = \psi^{(N-2)}_T \{\psi^{(N-1)} T Q - \psi^{(N-1)} T Q \Delta H^{(N-1)} + Q\}$$

$$- \psi^{(N-2)}_T \{\psi^{(N-1)} T Q \psi^{(N-1)} \Delta H^{(N-2)} + Q \Delta H^{(N-2)}\} + Q$$

$$K^{(N-2)} = \psi^{(N-2)}_T K^{(N-1)} - \psi^{(N-2)}_T P^{(1)} \Delta H^{(N-2)} + Q$$

Equation (36) is now written

$$-\frac{\partial x^T(N - 2)}{\partial m(N - 3)} (P^{(2)}) x(N - 2) + \frac{\partial x(N - 2)}{\partial m(N - 3)} K^{(N-2)} x^d = 0$$

Using the argument $(N - 2)$ in place of $N$ in Eq. (10) we can show that $\partial x^d(N - 2)/\partial m(N - 3) = \Delta^T$. We therefore express (41) as

$$-\Delta^T P^{(2)} x(N - 2) + \Delta^T K^{(N-2)} x^d = 0$$

$$-\Delta^T P^{(2)} [\Phi x(N - 3) + \Delta m(N - 3)] + \Delta^T K^{(N-2)} x^d = 0$$

$$m(N - 3) = -(\Delta^T P^{(2)} \Delta)^{-1} (\Delta^T P^{(2)} \Phi) x(N - 3) + (\Delta^T P^{(2)} \Delta)^{-1} \Delta^T K^{(N-2)} x^d$$

Let

$$C^{(N-3)} = -(\Delta^T P^{(2)} \Delta)^{-1} (\Delta^T P^{(2)} \Phi)$$

and

$$H^{(N-3)} = (\Delta^T P^{(2)} \Delta)^{-1} \Delta^T K^{(N-2)}$$

then

$$m(N - 3) = C^{(N-3)} x(N - 3) + H^{(N-3)} x^d$$

$$x(N - 2) = [\Phi + \Delta C^{(N-3)}] x(N - 3) + \Delta H^{(N-3)} x^d$$
Let
\[ \psi^{(N-3)} = \Phi + \Delta C^{(N-3)} \quad (49) \]
\[ x(N - 2) = \psi^{(N-3)} x^{(N-3)} + \Delta H^{(N-3)} x^d \quad (50) \]

Continuing the development an iteration pattern develops for the \( P^{(i)}, Q^{(i)}, H^{(i)}, K^{(i)}, C^{(i)} \) coefficients:
\[ P^{(i)} = \psi^{(N-i)} T P^{(i-1)} \psi^{(N-i)} + Q \quad (51) \]
\[ C^{(N-(i+1))} = -(\Delta T P^{(i)} \Delta)^{-1} \Delta T P^{(i)} \Phi \quad (52) \]
\[ \psi^{(N-(i+1))} = \Phi + \Delta C^{(N-(i+1))} \quad (53) \]
\[ H^{(N-(i+1))} = (\Delta T P^{(i)} \Delta)^{-1} \Delta T K^{(N-i)} \quad (54) \]
\[ K^{(N-i)} = \psi^{(N-i)} T K^{(N-(i+1))} - \psi^{(N-i)} T P^{(i-1)} \Delta H^{(N-i)} + Q \quad (55) \]

where
\[ P^{(0)} = Q, \quad \psi^{(N)} = 0, \quad K^{(N)} = Q \quad (56) \]

The iteration pattern can be used to compute the sequence of control vectors by
\[ m(N - i) = C^{(N-i)} x(N - i) + H^{(N-i)} x^d \quad (57) \]

For \( N \rightarrow \infty \), the matrix \( C^{(N-i)} \) converges to a matrix \( C \) and the matrix \( H^{(N-i)} \) converges to a matrix \( H \). The control of the process can be carried out by using successively Eq. (18) of Section 9 and Eq. (57) of this section.

12. Absorber Example

Lapidus et al. gives a numerical example of this technique for the transient behavior of an absorption tower.

The nomenclature is defined as follows:
\[ x_s = \text{composition of liquid leaving the sth plate (vector)} \]
\[ y_s = \text{composition of vapor leaving the sth plate (vector)} \]
\[ S = \text{number of plates} \]
\[ H_v = \text{vapor hold-up on any plate} \]
\[ h = \text{liquid hold-up on any plate} \]
\[ L = \text{absorbent rate} \]
\[ G = \text{inert gas rate} \]
A material balance around the $s$th plate gives

$$L(x_{s+1} - x_s) + G(y_{s+1} - y_s) = H \frac{dy_s}{dt} + h \frac{dx_s}{dt}$$  \hspace{1cm} (1)

The vectors $x(s)$ and $y(s)$ are, of course, functions of time. Using a linear equilibrium relation, we have

$$y_s = \alpha x_s + \beta$$  \hspace{1cm} (2)

$$\frac{dx_s}{dt} = \frac{d}{e} x_{s-1} - \left( \frac{d + 1}{e} \right) x_s + \frac{1}{e} x_{s+1}, \quad s = 1, 2, ..., S$$  \hspace{1cm} (3)

where

$$d = \frac{L}{G\alpha}$$  \hspace{1cm} (4)

$$e = \frac{H\alpha + h}{G\alpha}$$  \hspace{1cm} (5)

The initial compositions on the plates are determined by

$$\frac{dx_s}{dt} = 0 \quad t = 0, \quad s = 1, 2, ..., S$$  \hspace{1cm} (6)

The initial feed conditions are

$$x_s(t) = x_0(t), \quad s = 0$$  \hspace{1cm} (7)

$$y_s(t) = y_{s+1}(t), \quad s = S + 1$$  \hspace{1cm} (8)

In order to put the control problem in the form of Eq. (1), Section 9, or its linearized counterpart Eq. (12), Section 9, we make two observations. The first is that Eqs. (1) and (2) are linear and it is not necessary to linearize the equations around an equilibrium point. This means that the equations we set up will be directly in terms of the original variables and not in terms of the deviations from equilibrium. The second observation is that we consider $x_0$ and $y_{S+1}$ to be the control or manipulated variables.

Writing out the vector matrix equation (3) for the set $(x_1, x_2, ..., x_s)$, we obtain

$$\frac{dx_1}{dt} = \frac{d}{e} x_0 - \frac{d + 1}{e} x_1 + \frac{1}{e} x_2$$

$$\frac{dx_2}{dt} = \frac{d}{e} x_1 - \frac{d + 1}{e} x_2 + \frac{1}{e} x_3$$  \hspace{1cm} (9)
\[
\frac{dx_3}{dt} = \frac{d}{e} x_2 - \frac{d + 1}{e} x_3 + \frac{1}{e} x_4 \\
\vdots \\
\frac{dx_S}{dt} = \frac{d}{e} x_S - \frac{d + 1}{e} x_S + \frac{1}{e} \frac{y_{S+1} - \beta}{\alpha}
\]

where \(x_{S+1}\) is replaced by \((y_{S+1} - \beta)/\alpha\) from (2).

By rearranging (9), we may write it as a sum of two terms, the free and the forcing vectors:

\[
\frac{dx}{dt} = Ax + Dm
\]

where

\[
A = \begin{bmatrix}
\frac{-d + 1}{e} & \frac{1}{e} \\
\frac{d}{e} & -\frac{d + 1}{e} & \frac{1}{e} \\
\vdots & \vdots & \vdots \\
\frac{d}{e} & -\frac{d + 1}{e} & \frac{1}{e} \\
\frac{d}{e} & -\frac{d + 1}{e} & \frac{1}{e}
\end{bmatrix}
\]

\[
x = \begin{pmatrix}
x_1 \\
x_2 \\
\vdots \\
x_S
\end{pmatrix}
\]

\[
D = \begin{bmatrix}
d \\
\frac{d}{e} & 0 \\
0 & 0 \\
0 & 0 \\
\vdots & \vdots \\
0 & \frac{1}{e}
\end{bmatrix}
\]

\[
m = \begin{pmatrix}
x_0 \\
\frac{y_{S+1} - \beta}{\alpha}
\end{pmatrix}
\]
13. Numerical Evaluation of Absorber Example

For the theory developed in Sections 9–11 and for the absorber described in Section 12, we desire to minimize over \( N \) stages of time the function

\[
J(m(k-1)) = \sum_{k=1}^{N} [x^d - x(k)]^T Q [x^d - x(k)]
\]

(1)

where

\[
Q = I, \quad \text{identity matrix}
\]

(2)

The specific case we consider is the absorber described by the following data:

- \( S = 6 \) plates
- \( \alpha = 0.72 \)
- \( \beta = 0 \)
- \( L = 40.8 \) lb/min
- \( G = 66.7 \) lb/min
- \( H_v = 1.0 \)
- \( h = 75 \)

From these we find

\[
d = 0.850000; \quad 1/e = 0.634115
\]

Initially, the absorber is in equilibrium with the inlet liquid composition \( x_0 = 0 \) (pure liquid) and the inlet vapor composition

\[
y_{S+1} = 0.2 \frac{\text{lb solute}}{\text{lb inert}}
\]

At time, \( t = 0 \), the inlet vapor composition is stepped up to \( y_{S+1} = 0.3 \), while the inlet liquid composition \( x_0 = 0 \).

We desire to drive the process in \( N \) time stages from the initial conditions corresponding to \( x_0 = 0, y_{S+1} = 0.2 \) to the final or desired equilibrium conditions corresponding to \( x_0 = 0, y_{S+1} = 0.3 \). In addition, we wish to compare the dynamic programming solution time response with the normal transient response to the step input.

To find the initial state tray compositions, we set in Eq. (3), Section 12, \( dx_s/dt = 0, s = 1, 2, ..., S, \) with \( x_0 = 0, y_{S+1} = 0.2 \). Similarly, the final steady-state solution is found at \( dx_s/dt = 0 \) for \( x_0 = 0 \) and \( y_{S+1} = 0.3 \). The initial and final desired tray compositions are given in Table 1.
The solution of Eq. (10) of Section 12 is given by

\[ x(k + 1) = \Phi x(k) + \Delta m(k) \]  

(3)

For a sampling period of \( \tau = 1 \) minute, the matrices \( \Phi \) and \( \Delta \) are readily evaluated

\[ \Phi = \sum_{i=1}^{20} \frac{(A \tau)^i}{i!} \]

(4)

\[ \Delta = \int_0^\tau \Phi(\tau - \lambda) d\lambda D \]

(5)

The optimal control vectors are defined by

\[ m(k) = Cx^{(k)} + Hx^d \]

(6)
For $\tau = 1$ minute, $C^{(N-i)}$ and $H^{(N-i)}$ converge after 16 iterations to $C$ and $H$

$$C = \begin{bmatrix}
-1.22763 & -1.13746 & -0.69774 & -0.37809 & -0.16869 & -0.03888 \\
-0.03000 & -0.14597 & -0.33898 & -0.61705 & -0.98475 & -1.09026
\end{bmatrix}$$

(7)

$$H = \begin{bmatrix}
2.18511 & 1.49812 & 0.91418 & 0.41783 & 0.00406 & 0.36268 \\
-0.37349 & 0.14998 & 0.59493 & 0.97315 & 1.29463 & 1.56780
\end{bmatrix}$$

(8)

Starting with $x(0)$, the initial state vector and $x^d$, the desired vector as given in Table 1, Eq. (6) is used to generate the manipulated vectors $m(0)$. The computed value $m(0)$ is substituted into Eq. (3) in order to calculate $x(1)$. The value $m(1)$ is found from Eq. (6), using $x(1)$, and the $x(2)$ is found from Eq. (3). In this manner the response of the system to meet the minimization of

$$\sum_{k=1}^{N} [x^d - x(k)]^T Q [x^d - x(k)]$$

is found.

**TABLE 2**

**VARIATION IN CONTROL VALUES AND RESPONSE AS FUNCTION OF SAMPLING TIME**

<table>
<thead>
<tr>
<th>Sampling times</th>
<th>Control values</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$x_0$</td>
<td>$(y_\tau - \beta)/\alpha$</td>
</tr>
<tr>
<td>0$\tau$</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1$\tau$</td>
<td>0.21828</td>
<td>0.76370</td>
</tr>
<tr>
<td>2$\tau$</td>
<td>0.08949</td>
<td>0.49771</td>
</tr>
<tr>
<td>3$\tau$</td>
<td>0.06414</td>
<td>0.46560</td>
</tr>
<tr>
<td>4$\tau$</td>
<td>0.04589</td>
<td>0.44826</td>
</tr>
<tr>
<td>5$\tau$</td>
<td>0.03242</td>
<td>0.43771</td>
</tr>
<tr>
<td>6$\tau$</td>
<td>0.02267</td>
<td>0.43090</td>
</tr>
<tr>
<td>7$\tau$</td>
<td>0.01574</td>
<td>0.42637</td>
</tr>
<tr>
<td>8$\tau$</td>
<td>0.01089</td>
<td>0.42330</td>
</tr>
<tr>
<td>9$\tau$</td>
<td>0.00751</td>
<td>0.42122</td>
</tr>
<tr>
<td>10$\tau$</td>
<td>0.00518</td>
<td>0.41979</td>
</tr>
<tr>
<td>20$\tau$</td>
<td>0.00012</td>
<td>0.41674</td>
</tr>
<tr>
<td>$\infty$$\tau$</td>
<td>0.00000</td>
<td>0.41666</td>
</tr>
</tbody>
</table>

Table 2 lists the values of the control variables and the response of $x_0$. Figure 5 shows the dynamic programming response and the normal
NUMERICAL EVALUATION OF ABSORBER EXAMPLE

Fig. 5. Exit liquid composition versus time.

- Initial value of 0.2546
- Dynamic programming response
- Sampling period = 1 minute
- Final equilibrium of 0.3820

\[ x^{9}_{L}\text{Exit liquid composition} \]
transient response. It is quite apparent that the dynamic programming technique drives the process to the desired final state considerably faster than the normal transient response.

14. Alternate Method of Kalman

Using somewhat different notation but basically the same ideas, Kalman has presented another very similar method of generating the optimal control sequence for the linearized matrix differential equations with quadratic performance index.

The basic set of linear differential equations is expressed in vector notation

$$\frac{dx}{dt} = Ax + Dm$$

where the terms are defined in Section 9.

The solution in the time domain is given by

$$x(t) = e^{At} x(t_0) + \int_{t_0}^{t} e^{A(t-\lambda)} Dm d\lambda$$

Assuming that the input or manipulated variables can be changed only at discrete intervals \( t_0 \), \( t_0 + \tau \), \( t_0 + 2\tau \), etc., and are held constant between the sampling periods, Eq. (2) may be written as

$$x(t_0 + \tau) = \Phi(\tau) x(t_0) + \Delta(\tau) m(t_0)$$

where

$$\Phi(\tau) = e^{A\tau}$$

$$\Delta(\tau) = \int_{t_0}^{t_0+\tau} e^{A(t-\lambda)} Dd\lambda = \int_{0}^{\tau} (\Phi(\tau - \lambda)) Dd\lambda$$

The performance index \( P_N(x(t_0)) \) is given by

$$P_N(x(t_0)) = \sum_{k=1}^{N} [x^d - x(t_0 + k\tau)]^T Q [x^d - x(t_0 + k\tau)]$$

We define the minimum of the performance index†

$$P_{N+1}^\circ(x(t_0)) = \min_{m(t_0)} P_N(x(t_0))$$

† In order to establish a correspondence between Kalman’s notation in this section and the notation in Sections 9–11, let us note

\( x(t_0) \sim x(0); \quad x(t_0 + k\tau) \sim x(k) \)

\( P_N(x(t_0 + k\tau)) \sim J(m(k)); \quad P_N^\circ(x(t_0 + k\tau)) \sim F_N(x) \)
Using the Principle of Optimality the functional equations are

\[ P_{N+1}^0(x(t_0)) = \min_{m(t_0)} \{ [x^d - x(t_0 + \tau)]^T Q [x^d - x(t_0 + \tau)] + P_N^0 x(t_0 + \tau) \} \]  

(8)

\[ P_1^0(x(t_0)) = \min_{m(t_0)} \{ [x^d - x(t_0)]^T Q [x^d - x(t_0)] \} \]  

(9)

Kalman states that by induction the optimal performance index has the form

\[ P_N^0(x(t_0)) = x^T(t_0) U_N x(t_0) - 2 x^T(t_0) R_N x^d + (x^d)^T S_N x^d \]  

(10)

where \( U_N, R_N, S_N \) are \( n \times n \) and \( n \times m \), and \( m \times m \) matrices, and

\[ U_0 = R_0 = S_0 = 0 \]  

(11)

Equation (10) with the argument \( x(t_0 + \tau) \) may be substituted into Eq. (8)

\[ P_{N+1}^0(x(t_0)) = \min_{m(t_0)} \{ [x^d - x(t_0 + \tau)]^T Q [x^d - x(t_0 + \tau)] \} \]

\[ + x^T(t_0 + \tau) U_N x(t_0 + \tau) - 2 x^T(t_0 + \tau) R_N x^d + (x^d)^T S_N x^d \} \]  

(12)

Substituting (3) into (12), taking the derivative of the right-hand side of (12) with respect to \( m(t_0) \) and setting it equal to zero, we find

\[ \Delta^T (R_N + Q) x^d - \Delta^T (U_N + Q) \{ \Phi x(t_0) + \Delta m(t_0) \} = 0 \]  

(13)

The optimal control vector \( m^0(t_0) \) then is found by rearranging (13):

\[ m^0(t_0) = B_N x^d - A_N x(t_0) \]  

(14)

where

\[ A_N = [\Delta^T (U_N + Q) \Delta]^{-1} \Delta^T [U_N + Q] \Phi \]  

(15)

\[ B_N = [\Delta^T (U_N + Q) \Delta]^{-1} \Delta^T (R_N + Q) \]  

(16)

If we substitute (14) into (3), we have

\[ x(t_0 + \tau) = (\Phi - \Delta A_N) x(t_0) + \Delta B_N x^d \]  

(17)

Substituting (17) in (12) we write

\[ P_{N+1}^0(x(t_0)) \]

\[ = \{ [\Phi - \Delta A_N] x(t_0) + \Delta B_N x^d \}^T (U_N + Q) [\Phi - \Delta A_N] x(t_0) + \Delta B_N x^d \}

\[- 2 [\Phi - \Delta A_N] x(t_0) + \Delta B_N x^d \}^T (Q + R_N) x^d + (x^d)^T (S_N + Q) x^d \} \]

(18)
which after collecting terms appears as

$$
\mathcal{P}_{n+1}(x(t_0)) = \left( [(\Phi - \Delta A_N) x(t_0)]^T (U_N + Q) (\Phi - \Delta A_N) x(t_0) + 2 [\Delta B_N x^d]^T (U_N + Q) [\Phi - \Delta A_N] x(t_0) - 2 [\Phi - \Delta A_N] x(t_0)]^T [Q + R_N] x^d + [\Delta B_N x^d]^T (U_N + Q) [\Delta B_N x^d] - 2 [\Delta B_N x^d]^T [Q + R_N] x^d + (x^d)^T (S_N + Q) x^d \right)
$$

Another way to express \( \mathcal{P}_{n+1}(x(t_0)) \) is by way of (10)

$$
\mathcal{P}_{n+1}(x(t_0)) = x^T(t_0) U_{n+1} x(t_0) - 2 x^T(t_0) R_{n+1} x^d + (x^d)^T S_{n+1} x^d
$$

Comparing the two expressions for \( \mathcal{P}_{n+1}(x(t_0)) \), Eqs. (19) and (20), for the matrixes in brackets of \( x^T(t_0) \) \( x(t_0) \), \( x^T(t_0) \) \( x^d \), and \( (x^d)^T \) \( x^d \), we find the recursion relationships

$$
U_{n+1} = (\Phi - \Delta A_N)^T (U_N + Q) [\Phi - \Delta A_N] \tag{21}
$$

$$
S_{n+1} = (\Delta B_N)^T [(U_N + Q) \Delta B_N - 2 (Q + R_N)] + (S_N + Q) \tag{22}
$$

$$
R_{n+1} = (\Phi - \Delta A_N)^T [(Q + R_N) - (U_N + Q) \Delta B_N] \tag{23}
$$

where by Eq. (11)

$$
U_0 = R_0 = S_0 = 0
$$

As \( N \to \infty \), the terms \( A_N \) and \( B_N \) converge to fixed numbers and the sequence of control actions may be found from Eqs. (3) and (14).

The procedure developed here is essentially the same as the optimization procedure developed in Sections 9–11. In the previous sections the optimization was carried out directly on the quadratic form and from it the recursion formulas (51)–(57) in Section 11 were developed. In this section, advantage was taken of the expanded quadratic form of Eq. (10) from which the recursion formulas were developed.

If we compare Eq. (57) of Section 11, and Eq. (14) of this section, we observe that \( B_N \) corresponds to \( H^{(N-i)} \) and \( -A_N \) corresponds to \( C^{(N-i)} \).

### 15. Review of Kalman's Method

The method proposed by Kalman is to linearize a set of nonlinear differential equations around an equilibrium point. The solution to the
linear set of equations can be found. For a quadratic performance index, a set of recursion formulas is developed using the Principle of Optimality. The equations are developed over a discrete number of time stages where the control vector is held constant between stages but may vary at each time stage. As the number of stages increases, $N \to \infty$, the values of the recursion formulas converge to constant values. These are used to generate the values of the manipulated vector.

The method as presented here does not consider inequality constraints or limits on the size of incremental adjustments in the control vector from stage to stage.

REFERENCES


PROBLEMS

1. In reference [6] Lapidus et al. develop an expression comparable to Eq. (57), Section 11, namely,

$$m^a(k) = C x(k) + B x^d$$

where

$$C = \lim_{N \to \infty} C^{[N-(i+1)]} = -(\Delta T P^{(i)} \Delta T)^{-1} \Delta T P^{(i)} \Phi$$

$$B = (\rho_{ss} \rho_{ss}^{-1} \rho_{ss} Q)$$

$$\rho_{ss} = (I - \Phi)^{-1} \Delta$$

Reconcile $H^{(N-i)}$ in Eq. (57), as $N \to \infty$ with $B$ above.
\[ P_0(x(t_0)) = x^T(t_0) U_N x(t_0) - 2x^T(t_0) R_N x^d + (x^d)^T S_N x^d \]
develop
\[ m^0(t_0) = -A_N x(t_0) + B_N x^d \]
where
\[ A_N = [\Delta^T(U_N + Q) \Delta]^{-1} \Delta^T(U_N + Q) \Phi \]
\[ B_N = \Delta^T[(U_N + Q) \Delta]^{-1} \Delta^T(R_N + Q) \]
\[ U_{N+1} = (\Phi - \Delta A_N)^T (U_N + Q) \Phi \]
\[ R_{N+1} = (\Phi - \Delta A_N)^T (R_N + Q) \]
\[ S_{N+1} = S_N + Q - B_N^T \Delta^T(R_N + Q) \]
Show that Eqs. (21)-(23) for \( U_{N+1}, R_{N+1}, S_{N+1} \) of Section 14 reduce to those above.

3. Consider a continuous stirred tank reactor equipped with a heating coil. For the first-order nonisothermal exothermic reaction
\[ A \rightarrow B \]
the material and heat balances are
\[ \frac{dy_A}{dt} = \frac{u}{V} y_A y_{A0} - \frac{u}{V} y_A - k(T)y_A \]
\[ \frac{dT}{dt} = \frac{u}{V} T_0 - \frac{u}{V} T - k(T)y_A \frac{\Delta H}{\rho c} - \frac{UKF_c(T - T_{ce})}{V\rho c(1 + KF_c)} \]
where
- \( y_A \) = reactant concentration
- \( T \) = temperature of reaction mixture
- \( u \) = flow rate into and out of reactor
- \( V \) = reactor volume
- \( y_{A0} \) = feed concentration of reactant
- \( T_0 \) = feed temperature of reactant stream
- \( T_{ce} \) = cooling water inlet temperature
- \( U, K, \rho, c \) = heat parameters
- \( k(T) = \alpha e^{-E/RT} \)
- \( F_c \) = cooling water rate
- \( E \) = activation energy

Define
\[ y_A - y_A^* = x_1, \quad T - T^* = x_2, \quad F_2 - F_2^* = m_1 \]
where the asterisks refer to equilibrium values.
Show that the material balance equations may be expressed as

\[ \frac{dx}{dt} = Ax + Dm \]

where

\[ A = \begin{pmatrix} -\frac{u}{V} + k(T^*) & \frac{-k(T^*) \, Ey^*_A}{RT^*} \\ -\frac{\Delta H}{\rho c} k(T^*) - \frac{u}{V} + \left(\frac{\Delta H}{\rho c}\right) \frac{k(T^*) \, Ey^*_A}{RT^*} + \frac{UKF^*_2}{Vpc(1 + KF^*_2)} \end{pmatrix} \]

\[ D = \begin{pmatrix} 0 \\ \frac{UK(T^* - T_{ce})}{Vpc(1 + KF^*_2)^2} \end{pmatrix} \]

\[ x = \begin{pmatrix} y^*_A - y^*_A \\ T - T^* \end{pmatrix} \]

\[ m = \begin{pmatrix} 0 \\ F_2 - F^*_2 \end{pmatrix} \]

(Amundson, N. R., Aris, R., Kalman, R. E., and Lapidus, L. [1].)

4. Consider a train of six liquid-liquid countercurrent extraction units with a secondary feed to the third stage shown in the sketch below. The solvents are
considered immiscible and secondary feed is taken from two large reservoirs, each containing a constant composition (not the same in each reservoir) of solute. A single solute is to be extracted with $x$ and $y$ used to indicate extract and raffinate compositions, respectively. The $b$ and $w$ are the solvent flow rates; $x_{F1}$, $x_{F2}$, and $F_1$, $F_2$ are the solute compositions and solvent flow rates from reservoir 1 and 2, respectively, which combine to make a feed composition $x_F$ and flow rate $F$ to the third stage. If $h$ is the extract holdup and $H$ the raffinate holdup (assumed constant per stage), a material balance on each stage leads to the dynamic material balances

$$h \frac{dy_n}{dt} + H \frac{dx_n}{dt} = b[x_{n-1} - x_n] + w[y_{n+1} - y_n]; \quad n = 1, 2$$

$$h \frac{dy_n}{dt} + H \frac{dx_n}{dt} = bx_{n-1} - b^* x_n + w[y_{n+1} - y_n] + F x_F; \quad n = 3$$

$$h \frac{dy_n}{dt} + H \frac{dx_n}{dt} = b^*[x_{n-1} - x_n] + w[y_{n+1} - y_n]; \quad n = 4, 5, 6$$

where $b^* = b + F$. Assuming a linear equilibrium

$$y_n = \alpha x_n + \beta$$

and defining

$$d = \frac{b}{a}; \quad d^* = \frac{b^*}{a}; \quad p = \frac{w \alpha}{a}; \quad a = h \alpha + H$$

show that

$$\frac{dx}{dt} = F x + G_1 u_1 + G_2 u_2$$

where

$$w = \begin{bmatrix}
-(d+p) & p & \cdot & \cdot & \cdot \\
d & -(d+p) & p & \cdot & \cdot \\
\cdot & d & -(d^*+p) & p & \cdot \\
\cdot & \cdot & d & -(d^*+p) & p \\
\cdot & \cdot & \cdot & d^* & -(d^*+p) \\
\cdot & \cdot & \cdot & \cdot & d^* & -(d^*+p)
\end{bmatrix}$$

$$x = \begin{bmatrix}
x_1 \\
x_2 \\
\vdots \\
x_n
\end{bmatrix}; \quad G_1 = \begin{bmatrix}
d & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0
\end{bmatrix}; \quad G_2 = \begin{bmatrix}
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0
\end{bmatrix}; \quad u_1 = \begin{bmatrix}
x_0 \\
y_1 - \beta \\
\vdots \\
y_n
\end{bmatrix}; \quad u_2 = \begin{bmatrix}
x_F
\end{bmatrix}$$

(Amundson, N. R., Aris, R., Kalman, R. E., and Lapidus, L. [1].)
5. For the chemical reactor described in Section 6, develop the functional equations for the
\[
\min \max [C(L) - C_d]^2
\]

6. Consider a sequence of stirred reactors, shown in the sketch below in which it is possible to control the level in each reactor (and hence the holding time) and the reactor temperature.

Define
\[
\begin{align*}
  c_k &= \text{concentration leaving the } k\text{th reactor} \\
  \theta_k &= \text{holding time in the } k\text{th reactor} \\
  r_k &= r(c_k, T_k) = \text{reaction rate in } k\text{th reactor} \\
  T_k &= \text{temperature in the } k\text{th reactor}
\end{align*}
\]

The material balance equation is
\[
c_{k+1} = c_k - \theta_k r_k; \quad k = 1, 2, ..., N
\]

The temperature is constrained by
\[
T_{\text{Min}} \leq T_k \leq T_{\text{Max}}; \quad k = 1, 2, ..., N
\]

The total holding time is constrained by an upper limit:
\[
\sum_{k=1}^{N} \theta_k \leq M
\]

If we desire to maximize \(c_1\), show that the functional equations are
\[
\begin{align*}
  f_N(c_{N+1}, M) &= \max_{\theta_N, T_N} [\theta_N r_N + f_{N-1}(c_N, M - \theta_N)] \\
  f_1(c_2, M) &= \max_{\theta_1, T_1} [\theta_1 r_1]
\end{align*}
\]

Write an alternate formulation.


7. Consider the multi-stage process with recycle, shown in the sketch below.
Let

\[ q = \text{flow rate} \]
\[ P_i = \text{quality of process stream leaving the } ith \text{ stage} \]
\[ P_f = \text{quality of fresh feed stream} \]
\[ w_i = \text{operating variables, such as steam pressure or solvent addition rate} \]
\[ c(w_i) = \text{cost of operating the } ith \text{ stage} \]
\[ r = \text{recycle rate} \]
\[ R = \text{value of process stream} \]

Let us observe that the process stream quality is a function of the quality at the previous stage and the operating variables

\[ P_i = T(P_{i-1}, w_i) \]

The quality of the mixed stream that enters the first stage is

\[ P_0 = \frac{qP_f + rP_N}{q + r} \]

The objective is to maximize

\[ R[P_f - P_N]q - \sum_{i=1}^{N} c(w_i) \]

Show that the optimal conditions are found from

(a) \( f_N(P_f, q) = \max_{w_1, \ldots, w_N} [R(P_0 - P_N)(q + r) - \sum_{i=1}^{N} c(w_i)] \)

(b) Observe that the proper value for \( P_0 \) is found by an iterative process. Suggest a method for calculating \( P_0 \).

(c) Show that the sequence of trial policies \( P_0^{(0)}, P_0^{(1)}, P_0^{(1)} \) converges.


8. A batch reaction is described by

\[ \frac{dx_1}{dt} = -k_1 x_1 x_2 - 2k_2 x_2^2; \quad \frac{dx_2}{dt} = -k_1 x_1 x_2 \]

where

\[ k_1 = A_1 \exp \left( -\frac{E_1}{R\theta} \right); \quad k_2 = A_2 \exp \left( -\frac{E_2}{R\theta} \right) \]

where \( \theta \) is the temperature. We desire to control this reaction so that at the final time the quantity

\[ [x_2(T) - x_1(T)] \]
is a maximum. The initial conditions are
\[ x_1(0) = c_1; \quad x_2(0) = c_2 \]

Set up the functional equations.

9. For a series of stirred tank reactors, the kinetics of the reaction \( A \to B \to C \) are described by

\[ x_{k+1} = (1 + a_k)x_k; \quad y_{k+1} = -a_kx_k + (1 + b_y)y_k \]

where

- \( x_k = \) the concentration of \( A \) at the \( k \)th reactor
- \( y_k = \) the concentration of \( B \) at the \( k \)th reactor
- \( a_k, b_y \) are known functions of temperature and the holding time at the \( k \)th reactor

If the objective is to maximize \([y_1 - y_{N+1}]\), show that the functional equations take the following forms due to the homogeneity of the equations:

\[ f_N(x_{N+1}, y_{N+1}) = \max_{a_N, b_y} \sum_{k=1}^{N} (y_k - y_{k+1}) \]

\[ f_N(x_{N+1}, y_{N+1}) = x_{N+1}g_N(x_{N+1}) \]

\[ f_N(x_{N+1}, y_{N+1}) = \max_{a_N, b_y} [y_N - y_{N+1} + f_{N-1}(x_N, y_N)] \]

or

\[ f_N(x_{N+1}, y_{N+1}) = x_{N+1} \max_{a_N, b_y} \left[ \frac{a_N}{(1 + a_N)(1 + b_y)} - \left( \frac{b_N}{1 + b_N} \right) z_{N+1} \right. \]

\[ \left. + \frac{1}{1 + a_N} g_{N-1}(x_N) \right] \]


10. Given a batch reactor process described by

\[ \frac{dc}{dt} = -r(c, T); \quad \frac{dT}{dt} = -Hr(c, T) + Q \]

where

- \( c = \) concentration
- \( H = \) heat of reaction
- \( r = \) reaction rate
- \( Q = \) heat removed
- \( T = \) temperature
It is desired to control the reaction from an initial concentration \( c_0 \) and temperature \( T_0 \) to a final concentration \( c_1 \) by adjusting \( Q \). If we define

\[
f(c_0, T_0) = \text{minimum time required to drive the reaction from the initial state } (c_0, T_0) \text{ to the final concentration } c_1
\]

show that control equation is given by

\[
\text{Min } [1 + f_\sigma(c_0, T_0) + f_T(Hr(c_0, T_0) - Q)] = 0
\]

Show that the optimal policy is to maximize \( Q \) if \( f_T > 0 \), and to minimize \( Q \) if \( f_T < 0 \). What is the importance of \( f_T = 0 \)?


11. A tank of constant cross-sectional area \( A \) and liquid level \( h \) is filled by a flow rate \( Q(t) \) and drained simultaneously through an orifice at the bottom of the tank. The differential equation of the processes is

\[
A \frac{dh(t)}{dt} = Q(t) - c_d \sqrt{2gh(t)}
\]  
(1)

where \( c_d \) is the orifice coefficient.

(a) If we consider \( Q(t) \) to consist of an average flow \( Q_{i \text{ avg}} \) plus a fluctuating flow \( AQ(t) \), show that response of the system is given by

\[
- \frac{t}{A} = \frac{2}{k} \left( \sqrt{h} - \frac{Q_{i \text{ avg}}}{k} \right) + \frac{2(Q_{i \text{ avg}} + AQ_i)}{k^2} \ln \left( \frac{Q_{i \text{ avg}} + AQ_i - k \sqrt{h}}{AQ_i} \right)
\]
(2)

where

\[
Q_{i \text{ avg}} = \text{initial value of average flow } = Q_{i \text{ avg}} \text{ at time } = 0
\]
\[k = c_d \sqrt{2g}\]
\[AQ_i = \text{fluctuating component of flow}\]

Find the steady state value for a unit step in \( AQ_i \).

(b) Let \( h = h_0 + z \), \( z < < h_0 \), where

\[
h_0 = \text{average head } = (Q_{i \text{ avg}})^2/k^2
\]
\[z = \text{variation in head around the average}\]

Linearize the equation to

\[
A \frac{dz}{dt} + \frac{kz}{2 \sqrt{h_0}} = AQ_i(t)
\]
(3)
Develop the time response and the steady-state value for a unit change in $\Delta Q_z(t)$.

(c) Set up the functional equations of dynamic programming to drive the head from $h_0$ at $t = 0$ to the steady-state value in minimum time for a unit charge in $\Delta Q_z(t)$.

(d) Evaluate the functional equations and compare the dynamic programming solution to the solutions for parts (a) and (b).

1. Introduction

Inventory problems pose a challenging field of application for dynamic programming. Some inroads have been made in this area but a great deal of work still needs to be done.

The inventory problem often is expressed as the minimization of cost for an operation involving the ordering of a commodity, the storing of the commodity, and the selling of the commodity. The costs normally include the ordering, storing, and selling charges. In addition, there may be red tape or fixed charges, penalty charges for failure to meet sales demands, and premium prices for fast order service. Inventory problems may be presented as continuous or discrete problems, with or without time lag in the execution of the ordering, storing, and selling functions.

Inventory problems may appear in deterministic or probabilistic form. For example, in a deterministic case, the purchase, storing, and selling prices, as well as the demand pattern, may be completely known. For the stochastic case, the price structure and/or the demand may be probabilistic variables.

The inventory problem gives rise to unusual problems in the calculus of variations. In treatments of this subject by various authors, the formal approach of the calculus of variations is seldom employed. In various formulations of the inventory problem, various investigators give the operating philosophy without actually solving the problem;
that is, the solution is given in policy space in terms of general principles. One familiar dictum is: if \( y \leq x^* \), order \((y - x^*)\) and if \( y \geq x^* \), do not order. Here \( y \) is the actual inventory level and \( x^* \) is the optimal level.

In this chapter, we present a warehouse problem and solve it by three different means. The first approach yields the familiar functional equations. The second approach provides a computational algorithm which is based on the linearity of the problem. The third approach develops an analytical solution from which the policy structure is determined and from which a computational procedure is generated. A continuous inventory problem with a stochastic demand pattern is discussed next. The formulation develops the optimal policy structure. A variation of the continuous inventory problem, a problem with setup charges, is given next to point out the difficulties that result from the inclusion of this one additional term. A finite time period counterpart to the continuous inventory problem is also given. Having discussed a problem with stochastic demand upon the inventory, we next present an inventory problem with the supply to the inventory, a stochastic variable. The purchasing of material on a fluctuating price market with a fluctuating usage pattern of the material is considered next. A linear production and inventory control model is developed which serves as the basis for three related problems; the least cost, the maximum profit, and the maximum rate of return on investment. A typical inplant inventory situation of temporary storage for the feed material and the product material is handled by dynamic programming. Finally, a general smoothing problem is set up. Following the same general approach, analytical results are given for a quadratic form of the discrete smoothing problem.

2. Warehousing Problem

We will set up the warehousing problem and solve it by three different dynamic programming methods. The first method sets up the functional equations and requires a search technique to develop the answers. The second method takes advantage of the linearity of the problem and uses this to develop a simple computational algorithm. The third method provides an analytical solution to the problem. From it the structure of the solution is deduced. This in turn leads to a computational solution.

The warehousing problem is stated as follows:

Given a warehouse with a known fixed capacity \( B \) and a known initial stock of a single commodity, which is subject to known seasonal selling price and purchase cost variations, what is the optimal pattern of purchasing, storing, and selling?
We consider the warehousing problem for \( N \) periods of time. We define for the \( i \)th time period:

- \( c_i \) = purchase cost per unit, known value
- \( p_i \) = selling price per unit, known value
- \( x_i \) = the quantity bought
- \( y_i \) = the quantity sold
- \( v \) = the initial inventory at the beginning of time stage \( i \)
- \( B \) = the fixed warehouse capacity

We desire to maximize the profit over \( N \) stages of time:

\[
P = \sum_{i=1}^{N} (p_i y_i - c_i x_i)
\]  

by the proper choice of \( x_i \) and \( y_i \) over the \( N \) stages.

The system is constrained by:

(a) Buying constraints: The stock on hand at the end of the \( i \)th period cannot exceed the warehouse capacity.

(b) Selling constraints: The amount sold in the \( i \)th period cannot exceed the amount available at the end of the period \((i - 1)\).

(c) Non-negativity constraints: The amounts purchased or sold in any period are non-negative.

The buying constraint may be expressed as

\[
v + (x_i - y_i) \leq B, \quad i = 1, 2, ..., N
\]  

The selling constraint is given by

\[
y_i \leq v, \quad i = 1, 2, ..., N
\]  

The non-negativity constraint states that

\[
x_i, y_i \geq 0
\]  

Let us define

\[
f_N(v) = \text{the maximum profit over an } N\text{-stage process starting with an initial inventory } v, \text{ subject to the constraints, Eqs. (2)-(4), and using an optimal policy}
\]

The functional equations are

\[
f_N(v) = \max_{x_N, y_N} \left[ p_N y_N - c_N x_N + f_{N-1}(v + x_N - y_N) \right]
\]
where

\[ v + x_N - y_N = \text{the new inventory} \quad (6) \]

\[ f_1(v) = \text{Max}_{x_N, y_N} \left[ p_N y_N - c_N x_N \right] = p_N v \quad (7) \]

Equations (5) and (7) together with (2)-(4), are perfectly adequate to solve the warehousing problem. The first method of solution requires searching for the proper values of \( x_i \) and \( y_i \) at each stage to achieve the maximization.

The second method of solution carries the first method one step further. If we observe that the objective function, the profit equation, is linear in \( x_N \) and \( y_N \) and if we observe that the constraints (2)-(4) are linear, we have a completely linear system. This system might be solved by linear programming, but we will continue with the second dynamic programming approach. We may take advantage of the linearity of the problem to reduce the effort to find the proper values for the \( x_i \) and \( y_i \).

If we use the constraints for stage \( N \), namely,

\[ y_N \leq v \quad (8) \]
\[ x_N - y_N \leq B - v \quad (9) \]
\[ x_N, y_N \geq 0 \quad (10) \]

we may delineate the solution space in Fig. 1.

In linear programming problems it is well known that the solution must exist at one of the vertices of the closed polyhedral. In Fig. 1 there are four possible vertices. These are: \((0, 0)\), \((B - v, 0)\), \((B, v)\), \((0, v)\)

Using the coordinates of these four points and Eqs. (5) and (6), we may write the expression for \( f_N(v) \) as

\[
 f_N(v) = \text{Max} \left\{ \begin{array}{l}
 f_{N-1}(v), \\
 -c_N(B - v) + f_{N-1}(B), \\
 p_N v + f_{N-1}(0), \\
 p_N v - c_N B + f_{N-1}(B)
 \end{array} \right\} 
 \quad (11)
\]

Equation (11) states that the maximum return is found by evaluating all four expressions and choosing the largest.

To show how we obtained these expressions, let us consider the point \( x_N = B - v, y_N = 0 \). For this point, the new inventory according to Eq. (6) is

\[ v + (B - v) - 0 = B \quad (12) \]
By (5) we have therefore

\[ f_N(v) = -c_N(B - v) + f_{N-1}(B) \]  

(13)

This is given as the second alternative in Eq. (11). In a similar manner, the return for the other vertices may be generated.

Since we know \( v, B, c_N, \) and \( p_N, \) we can easily evaluate Eq. (11) without any search procedure for the proper \( x_N \) and \( y_N. \) This is the principle advantage of the second dynamic programming approach.

![Solution space, warehousing problem.](image)

**Fig. 1.** Solution space, warehousing problem.

3. Warehousing Problem, Analytical Solution

In this section we present the third dynamic programming approach to the warehousing problem given in Section 2. Here we will develop an analytical solution that yields the policy structure as well as a working
computational procedure. The nomenclature is the same as that employed in Section 2.

From Section 2, the functional equation for the return is

\[ f_N(v) = \max_{x_N, y_N} \left[ p_N y_N - c_N x_N + f_{N-1}(u + x_N - y_N) \right], \quad N = 1, 2, \ldots \]  

(1)

where \( f_0(v) = 0 \) and where the maximum is taken over the region

\[
\begin{align*}
y_N &\leq v \\
v + x_N - y_N &\leq B \\
x_N, y_N &\geq 0
\end{align*}
\]

The analytical structure of the solution will be deduced from the above equation.

Let

\[ u = \text{the level of inventory at the end of the period under investigation} \]  

(5)

\[ u = v + x_N - y_N \]  

(6)

\[ u \leq B \]  

(7)

Using Eqs. (6) and (7), we express Eq. (1) as

\[ f_N(v) = \max_{0 \leq u \leq B} \left[ p_N y_N - c_N x_N + f_{N-1}(u) \right], \quad N = 1, 2, \ldots \]  

(8)

\[ f_N(v) = \max_{0 \leq u \leq B} \left[ \max_{x_N, y_N \geq 0} \left( p_N y_N - c_N x_N + f_{N-1}(u) \right) \right] 
\]

(9)

\[ f_N(v) = \max_{0 \leq u \leq B} \left[ \Phi_N(u, v) + f_{N-1}(u) \right], \quad N = 1, 2, \ldots \]  

(10)

The \( \Phi_N(u, v) \) is defined by

\[ \Phi_N(u, v) = \max_{x_N, y_N \geq 0} \left[ p_N y_N - c_N x_N \right] 
\]

(11)

The function \( \Phi_N(u, v) \) is a function of the initial inventory, \( v \), and the new inventory, \( u \). Referring to (11), we see that it is linear in \( x_N \) and \( y_N \). This means the maximization of the right-hand side of (11) can be achieved by investigating only the end points, namely, \( x_N = 0, y_N = 0, x_N = u, y_N = v \).
When the new inventory is equal to or less than the initial inventory, $0 \leq u \leq v$, this implies that some quantity is sold, so $y_N \neq 0$. The one end point considered is $x_N = 0, y_N = v - u$. This states that we buy nothing and sell the quantity $(v - u)$. The other end point considered is $x_N = u, y_N = v$. This states that we buy the entire new inventory $u$, and sell the initial inventory $v$.

For the region $0 \leq u \leq v$, Eq. (11) becomes

$$\Phi_N(u, v) = \max \left[ p_N(v - u), p_Nv - c_Nu \right]$$

Here, $\Phi_N(u, v)$ is taken as the larger of the two possibilities.

When the new inventory lies between the initial inventory and the warehouse capacity constraint, we have $v < u < B$. This implies that some quantity is bought, so $x_N \neq 0$. Again we consider two end points.

The one end point is $x_N = u - v, y_N = 0$. This states that we sell nothing and purchase $(u - v)$ units. The other end point is $x_N = u, y_N = v$, which states that we sell the entire initial inventory and purchase an entire new inventory.

For the region $v \leq u \leq B$

$$\Phi_N(u, v) = \max \left[ -c_N(u - v), p_Nv - c_Nu \right]$$

Again, $\Phi_N(u, v)$ is taken as the larger of the two possibilities.

Despite the linearity of $\Phi_N(u, v)$ the structure of $f_N(v)$ is not obvious. The following theorem has been established [15].

"The function $f_N(v)$ is linear in $v$, the coefficients being functions of $p_1, \ldots, p_N, c_1, \ldots, c_N$. The form of $f_N(v)$ is

$$f_N(v) = K_N(p_1, p_2, \ldots, p_N, c_1, c_2, \ldots, c_N) + L_N(p_1, p_2, \ldots, p_N, c_1, c_2, \ldots, c_N) v$$

Furthermore, the optimal policy $u$ is independent of $v$, the initial stock, and depends only upon the selling prices and costs."

We will now prove this by induction.

From Eq. (10), we know that

$$f_1(v) = \max \left[ p_1y_1 - c_1x_1 + f_0(u) \right]$$

The function $f_0(u)$ is identically zero since $f_N(v)$ is defined for only $N = 1, 2, \ldots$. Carrying out the maximization in Eq. (15), we find that

$$f_1(v) = p_1v$$

This states for a one-stage process, we sell the entire inventory $v$. 
We will assume first that
\[ f_{N-1}(v) = K_{N-1} + L_{N-1}v \] (17)
and show that this form holds for the \( N \)-stage process, namely,
\[ f_N(v) = K_N + L_Nv \] (18)

In Eqs. (17) and (18), the \( K_N, K_{N-1}, L_N, L_{N-1} \) are functions of the prices \( p_i \) and the costs \( c_i \).

We will consider first the case where \( c_N \geq p_N \). Due to the hypothesis concerning the linearity of \( f_{N-1}(u) \), the maximum must occur at \( u = 0 \), \( u = v \), or \( u = B \) in view of the two regions \( 0 \leq u \leq v \) and \( v \leq u \leq B \).

The optimum is expressed by one of the three possibilities:
\[ f_N(v) = \text{Max} [\Phi_N(0, v) + f_{N-1}(0), \Phi_N(v, v) + f_{N-1}(v), \Phi_N(B, v) + f_{N-1}(B)] \] (19)

We will first evaluate the \( \Phi_N \) functions. For \( c_N \geq p_N \) and \( u = 0 \), \( u = v \), \( u = B \), respectively, we have Eqs. (20), (22), and (24):

\[ \Phi_N(0, v) = \text{Max} [p_Nv, p_Nv] = p_Nv \] (20)

This states we buy nothing and sell the entire inventory \( v \).

\[ \Phi_N(v, v) = \text{Max} [p_N(v - v), p_Nv - c_NV] = \text{Max} [0, -(c_N - p_N)v] \] (21)
\[ \Phi_N(v, v) = 0 \] (22)

This corresponds to the condition of neither buying nor selling.

\[ \Phi_N(B, v) = \text{Max} [-c_N(B - v), p_Nv - c_NB] \] (23)
\[ \Phi_N(B, v) = -c_NB(B - v) \] (24)

This states that the quantity \( (B - v) \) is bought and nothing is sold. For the condition \( v = 0 \) and \( v = B \), Eq. (17) may be written as
\[ f_{N-1}(0) = K_{N-1} \] (25)
\[ f_{N-1}(B) = K_{N-1} + L_{N-1}B \] (26)

Into the right-hand side of Eq. (19) we now substitute Eqs. (20) and (25) for the first term; Eqs. (22) and (17) for the second term; and Eqs. (24) and (26) for the third term.
Equation (19) is now expressed as

\[
f_N(v) = \max \left[ p_N v_N + K_{N-1} + L_{N-1} v, -(c_N(B - v)) + K_{N-1} + L_{N-1} B \right]
\]  
(27)

The first term in Eq. (27) corresponds to buying nothing and selling the entire inventory \( v \). The second term corresponds to the condition of neither selling nor buying to maintain the initial inventory \( v \). The third term corresponds to the condition of selling nothing and buying \( (B - v) \).

We now examine the conditions for the choice of the largest term in Eq. (27). If the third term in Eq. (27) is greater than the second term

\[
-c_N(B - v) + K_{N-1} + L_{N-1} B \geq K_{N-1} + L_{N-1} v
\]  
(28)

\[
L_{N-1}(B - v) > c_N(B - v)
\]  
(29)

it follows that

\[
L_{N-1} > c_N
\]  
(30)

If the second term in Eq. (27) is greater than the first

\[
L_{N-1} v + K_{N-1} > p_N v + K_{N-1}
\]  
(31)

\[
L_{N-1} > p_N
\]  
(32)

Since \( L_{N-1} > c_N > p_N \), the third term is greater than the second, and the second is greater than the first. We conclude that these are conditions for the third term to be the largest, hence \( u = B \).

The second term is larger than the third term if \( L_{N-1} < c_N \). Therefore, the second term is largest when \( p_N < L_{N-1} < c_N \). In a similar manner, it can be shown that the first term in Eq. (27) is the largest if \( c_N > p_N \) and \( p_N > L_{N-1} \).

It is interesting to note that the maximizing \( u \) is independent of \( v \), the initial inventory.

If \( u = B \), then Eq. (27) becomes

\[
f_N(v) = -c_N(B - v) + K_{N-1} + L_{N-1} B 
\]  
(33)

\[
f_N(v) = (K_{N-1} + L_{N-1} B) - c_N v
\]  
(34)

If we let

\[
K_N = K_{N-1} + L_{N-1} B - c_N B
\]  
(35)

\[
L_N = c_N
\]  
(36)

we have

\[
f_N(v) = K_N + L_N v
\]  
(37)
If \( u = v \), Eq. (27) becomes
\[
 f_N(v) = K_{N-1} + L_{N-1}v
\]  
(38)

If we let
\[
 K_N = K_{N-1} \quad (39)
\]
\[
 L_N = L_{N-1} \quad (40)
\]
we write
\[
 f_N(v) = K_N + L_Nv \quad (41)
\]

If \( u = 0 \), Eq. (27) becomes
\[
 f_N(v) = p_Nv + K_{N-1} \quad (42)
\]

If we let
\[
 K_N = K_{N-1} \quad (43)
\]
\[
 L_N = p_N \quad (44)
\]
we write
\[
 f_N(v) = K_N + L_Nv \quad (45)
\]

In each case for \( u = 0, u = v, u = B \), Eq. (27) reduces to
\[
 f_N(v) = K_N + L_Nv \quad (46)
\]

where the \( K_N \) and \( L_N \) are functions of \( p_1, p_2, \ldots, p_N \) and \( c_1, c_2, \ldots, c_N \).

Starting with the statement for the one-stage process, Eq. (16) and the assumption for the \((N-1)\)-stage process, Eq. (17), we have executed the induction process for \( c_N > p_N \). By similar reasoning for the case of \( p_N > c_N \), the theorem can be proven. In Table 1 are listed the equations and conditions for the maximum for both \( c_N > p_N \) and \( p_N > c_N \).

The significance of the proof is threefold. First, it demonstrates the form of the conjectured function, namely, \( f_N(v) = K_N + L_Nv \). Second, it gives the structure of the policy. In particular, the policy takes the form of three different decisions. These include \( u = 0 \) which states that we deplete the inventory; \( u = v \), which states that the inventory level does not change; \( u = B \), which states that we increase the inventory to the warehouse limit. In addition, the proof yields equations and conditions summarized in Table 1, that may be used directly for computation.
### TABLE 1

**SUMMARY OF CONDITIONS FOR A MAXIMUM**

<table>
<thead>
<tr>
<th>Condition</th>
<th>Expression</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>$u = 0$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$c_N &gt; p_N$</td>
<td>$c_N &gt; p_N$</td>
<td>$u = v$</td>
</tr>
<tr>
<td>$p_N &gt; L_{N-1}$</td>
<td>$p_N &lt; L_{N-1} &lt; c_N$</td>
<td>$u = B$</td>
</tr>
<tr>
<td>$f_N(v) = K_{N-1} + p_N v$</td>
<td>$f_N(v) = K_{N-1} + L_{N-1} v$</td>
<td>(42)</td>
</tr>
<tr>
<td>$K_N = K_{N-1}$</td>
<td>$K_N = K_{N-1}$</td>
<td>(38)</td>
</tr>
<tr>
<td>$L_N = p_N$</td>
<td>$L_N = L_{N-1}$</td>
<td>(39)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Condition</th>
<th>Expression</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>$u = 0$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$p_N &gt; c_N$</td>
<td>$p_N &gt; c_N$</td>
<td>$u = B$</td>
</tr>
<tr>
<td>$L_{N-1} &gt; c_N$</td>
<td>$L_{N-1} &gt; c_N$</td>
<td>$u = B$</td>
</tr>
<tr>
<td>$f_N(v) = K_{N-1} + p_N v$</td>
<td>$f_N(v) = K_{N-1} + (L_{N-1} - c_N) B + c_N v$</td>
<td>(34)</td>
</tr>
<tr>
<td>$K_N = K_{N-1}$</td>
<td>$K_N = K_{N-1} + (L_{N-1} - c_N) B$</td>
<td>(35)</td>
</tr>
<tr>
<td>$L_N = p_N$</td>
<td>$L_N = L_{N-1}$</td>
<td>(40)</td>
</tr>
</tbody>
</table>

"3. WAREHOUSING PROBLEM, ANALYTICAL SOLUTION"
4. Numerical Example. Warehousing Problem

To illustrate the use of the theorem proved in Section 3, we give the following example. In Table 2, the first two columns are the given data, while the last three columns tabulate the results of the optimization. It is required to find for the warehousing problem the optimal return and policy for a 10-stage process.

To solve this problem, we must first determine the relative size of \( c_N \) with respect to \( p_N \), and \( L_{N-1} \) with respect to \( c_N \) and \( p_N \) using the data in Table 2.

**TABLE 2**

**Numerical Example**

<table>
<thead>
<tr>
<th>( N )</th>
<th>( c_i )</th>
<th>( p_i )</th>
<th>( K_i )</th>
<th>( L_i )</th>
<th>( u_i )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>3</td>
<td>0</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>1</td>
<td>( B^a )</td>
<td>2</td>
<td>( B )</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>5</td>
<td>( B )</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>5</td>
<td>( 3B )</td>
<td>3</td>
<td>( B )</td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>4</td>
<td>( 4B )</td>
<td>4</td>
<td>( B )</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>1</td>
<td>( 5B )</td>
<td>3</td>
<td>( B )</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>7</td>
<td>( 6B )</td>
<td>7</td>
<td>( B )</td>
</tr>
<tr>
<td>9</td>
<td>8</td>
<td>6</td>
<td>( 5B )</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>8</td>
<td>3</td>
<td>( 6B )</td>
<td>7</td>
<td>( v )</td>
</tr>
</tbody>
</table>

* \( B = \) the maximum \( u_i \).

The calculation procedure is described for the first three stages. For stage 1, we observe from Table 2 that \( c_1 > p_1(3 > 2) \), so we use the upper part of Table 1. Since \( K_0 = 0 \) and \( L_0 = 0 \), we see that \( p_1 > L_0 \) \((2 > 0)\). We, therefore, use the first column in the upper part of Table 1. This column corresponds to \( c_1 > p_1 \) and \( p_1 > L_0 \) and the policy \( u = 0 \). By Eq. (44), we have \( L_1 = p_1 = 2 \) and by Eq. (43), we have \( K_1 = K_0 = 0 \). The return by Eq. (42) is \( f_1(v) = p_1v = 2v \) and the policy is \( u = 0 \).

For stage 2, we observe from Table 2 that \( c_2 > p_2(5 > 3) \). In addition, from Table 2 and from the stage 1 calculation we observe that \( p_2 > L_1 \) \((3 > 2) \). As a result of \( c_2 > p_2 \) and \( p_2 > L_1 \), we must use the equations in column 1 of the upper part of Table 1. By Eq. (43) we note that \( K_2 = K_1 = 0 \), where \( K_1 \) is found in the stage 1 calculation. By Eq. (44), we have \( L_2 = p_2 = 3 \). The policy is \( u = 0 \) and the return is \( f_2(v) = 3v \).

Continuing for stage 3, we see from Table 2 that \( c_3 > p_3(2 > 1) \).
From the stage 2 calculation and from Table 2, we observe that \( L_2 > c_3(3 > 2) \). As a consequence of these inequalities, we use column 3 in the upper part of Table 1. By Eq. (35), we write \( K_3 = K_2 + (L_2 - c_3) B = B \) and by Eq. (36), \( L_3 = c_3 = 2 \). The return is \( f_3(v) = K_2 + (L_2 - c_3) B + c_3 v = B + 2v \) and the policy is \( u = B \).

Continuing in this manner, the last three columns of Table 2 are developed.

5. Comparison of the Warehousing Problem Solutions

The first approach yielded the typical dynamic programming functional equations. Their solution required a search procedure for the optimal values of the \( x_i \) and \( y_i \) over the \( N \) stages of time.

In the second approach we took advantage of the linearity of the problem and also of the functional equations of the first method. From the linearity we knew the optimal solution existed at one of the vertices of the simplex at each time stage. By means of the evaluation of the functional equations for each vertex, we determined the optimal return, namely, Eq. (11) of Section 2. This technique provided a workable easy-to-execute computational algorithm. It did not, however, give any insight into the structure of the optimal policy.

The third approach yielded an analytical solution which gave great insight into the structure of the optimal policy. In addition, it provided a convenient computational algorithm. In particular the third approach gave the following results.

1. The optimal \( N \)-stage policy is a linear function of the initial inventory with the coefficients dependent on the costs and selling prices.
2. The optimal policy at any stage is independent of the initial inventory at that stage.
3. The optimal policy has the following structure: do nothing for the first \( k \) stages \( (u = v) \), then oscillate between full \( (u = B) \) and empty \( (u = 0) \) warehouse for the remainder of the process.
4. The policy and return can be calculated readily from the information given in Table 1.

6. Continuous Inventory Model

It is often convenient in dynamic programming to consider continuous versions of the problems. In the inventory problem we will set up first
the continuous version of the fixed stock inventory problem and deduce from it an operating policy. To illustrate the technique a specific example will be worked out. From the continuous case we will set up the $N$-stage process analog.

We consider a single commodity stochastic inventory model. We desire to minimize the cost of ordering plus the penalty cost for failure to meet demands. The stochastic element enters in the demand requirement, whose probability density function is known. We will write first a general inventory equation and then we will specialize the ordering cost and penalty cost function to be linear in their arguments.

Let

\[ \varphi(s) \, ds = \text{the probability that the demand lies between } s \text{ and } s + ds \]

\[ K(s) = \text{the cost function of ordering items to increase the stock level} \]

\[ P(z) = \text{the cost penalty function of ordering items to meet an excess, } z, \text{ of demand over supply} \]

\[ a = \text{discount factor, the present worth factor} \]

\[ f(x) = \text{the expected minimum cost of satisfying the demand beginning with a supply } x \text{ (or level) using an optimal policy} \]

\[ y = \text{the inventory supply (or level) after the quantity } (y - x) \text{ is ordered} \]

The inventory cost consists of four terms

1. the cost of ordering \((y - x)\)
2. the penalty cost when the demand exceeds the supply \(y\)
3. the cost over the remainder of the time if the demand equals the new inventory \(y\)
4. the cost over the remainder of the time if the new supply \(y\) exceeds the demand.

The expected minimum cost is expressed as

\[
 f(x) = \min_{y > x} \left[ K(y - x) + a \int_y^\infty P(s - y) \varphi(s) \, ds + af(0) \int_0^\infty \varphi(s) \, ds \right. \\
 \left. + a \int_0^y f(y - s) \varphi(s) \, ds \right]
\]

The first term \(K(y - x)\) represents the cost of ordering the quantity \((y - x)\). The second term represents the discounted expected value of the penalty when the demand exceeds the inventory \(y\). The third term represents the discounted expected value of the minimum cost over the remaining time when the demand \(s\) equals the inventory \(y\). The last term represents the discounted expected cost over the remaining time when the supply exceeds the demand.
The terms in Eq. (1) may be better understood by referring to Fig. 2.

\[ K(y - x) = k(y - x) \]  

(1)

where \( k \) is a constant.

We specify here the cost of ordering is proportional to the size of the deficiency:

\[ P(s - y) = p(s - y) \]  

(2)

where \( p \) is a constant.

In order to develop computable results and to satisfy both mathematical and economic sense, we require the following conditions:

(a) \( k \) and \( p \) are positive constants. This follows since we desire positive costs for ordering and positive penalty costs. Negative ordering costs would mean we are paid by the vendor to purchase the commodity. Negative penalty costs of course vitiate the very concept of penalty.

(b) \( \varphi(s) > 0, \quad \int_0^\infty \varphi(s) \, ds = 1 \)
These follow from the definition of a probability density function

(c) \[ \int_{0}^{\infty} s \phi(s) \, ds \leq \infty \]

This states that the expected value of \( s \), \( E(s) = \int_{-\infty}^{\infty} s \phi(s) \, ds \) exists.

(d) \( 0 \leq a \leq 1 \), this states that the discount factor varies between zero and one.

(e) \( ap > k \). The justification for this will follow from the proof to be developed.

Before we continue this development let us digress for a moment to establish an important property. Let

\[ u(x) = \min_{y} v(x, y) \]  
(3)

where \( y \) is in a region \( R(x) \). If we assume that for an interval \( c \leq x \leq b \), that the minimum exists within the interval and that \( v \) is differentiable, then the value of \( y \) that determines \( u(x) \) is found from

\[ \frac{\partial v}{\partial y} = 0 \]  
(4)

Substituting the function \( y(x) \), (which may not be single valued) that gives the minimum in (4) into (3), we have

\[ u(x) = v(x, y) \]  
(5)

Within the region \( c \leq x \leq b \), we may write

\[ \frac{du}{dx} = \frac{\partial v}{\partial x} + \frac{\partial v}{\partial y} \frac{dy}{dx} \]  
(6)

But since \( \frac{\partial v}{\partial y} = 0 \) by (4), Eq. (6) reduces to

\[ \frac{du}{dx} = \frac{\partial v}{\partial x} \]  
(7)

This is the important property that we exploit below.

We rewrite Eq. (1) of Section 6 using the cost and penalty terms of (1) and (2) of this section:

\[ f(x) = \min_{\gamma \geq x} \left[ k\gamma - kx + a \int_{\gamma}^{\infty} (p\gamma - py) \phi(s) \, ds \right. \]
\[ \left. + af(0) \int_{\gamma}^{\infty} \phi(s) \, ds + a \int_{0}^{\gamma} f(y - s) \phi(s) \, ds \right] \]  
(8)
Differentiating the right-hand side of (8) with respect to \( y \), we obtain for \( y > x^+ \):

\[
k - ap \int_y^\infty \varphi(s) \, ds + a \int_0^y f'(y - s) \varphi(s) \, ds = 0 \tag{9}
\]

The values of \( y \) that satisfy (9) execute the minimization in (8). It is interesting to note the equation for minimum conditions does not contain \( x \), the original inventory level.

Using the results Eqs. (3)–(7) where \( f(x) \) corresponds to \( u(x) \) and the right-hand side of (8) corresponds to \( v(x, y) \) we write

\[
f'(x) = -k \tag{10}
\]

Since (10) is true for any initial stock \( x \), it is also true for the stock level \( (y - s) \)

\[
f'(y - s) = -k \tag{11}
\]

Using (11) in Eq. (9) for the last integral term, a simpler form is arrived at:

\[
k - ap \int_y^\infty \varphi(s) \, ds - ak \int_0^y \varphi(s) \, ds = 0 \tag{12}
\]

Recognizing with \( s \geq 0 \) that

\[
\int_0^\infty \varphi(s) \, ds = 1 \tag{13}
\]

by the definition of a probability function we may break the interval \((0, \infty)\) into two parts \((0, y)\) and \((y, \infty)\) and express (13) as

\[
\int_0^y \varphi(s) \, ds + \int_y^\infty \varphi(s) \, ds = 1 \tag{14}
\]

\(^{\dagger}\) Recalling the rule for differentiation under an integral sign

\[
\phi(\beta) = \int_{u_1(\beta)}^{u_2(\beta)} f(x, \beta) \, dx
\]

\[
\frac{d\phi(\beta)}{d\beta} = f(u_1, \beta) \frac{du_1}{d\beta} - f(u_2, \beta) \frac{du_2}{d\beta} + \int_{u_2(\beta)}^{u_1(\beta)} \frac{\partial f(x, \beta)}{\partial \beta} \, dx
\]
Substituting of (14) into (12) yields
\[ k - ap \left[ 1 - \int_0^y \varphi(s) \, ds \right] - ak \int_0^y \varphi(s) \, ds = 0 \]  \hspace{1cm} (15)
\[ \int_0^y \varphi(s) \, ds = \frac{ap - k}{a(p - k)} \]  \hspace{1cm} (16)

Equation (16) states there is value of \( y \) (expressed as the upper limit on the integral), that satisfies the right-hand side of the equation given the constants \( a, p, \) and \( k \). This value of \( y \) represents the value which minimizes the right-hand side of (8) under the listed assumptions. The fact that the \( y \) is unique, that there is only one value which executes this optimization, is guaranteed by \( \varphi(s) > 0 \) (assumption b). At the minimum the \( y = \bar{x} \), where \( \bar{x} \) is the unique root of Eq. (12).

Referring back to assumption (e), \( ap > k \), we observe in Eq. (16) that this condition must be met, so the value of the integral is positive which the definition of a density function requires.

We have shown that
\[ y = \bar{x}, \quad 0 \leq x \leq \bar{x} \]  \hspace{1cm} (17)

Carrying the development further, it can be shown that
\[ y = x, \quad x > \bar{x} \]  \hspace{1cm} (18)

To verbalize, the first policy, Eq. (17) states that when the stock level is below the known optimal level, we order up to the optimal level, namely, the quantity \((\bar{x} - x)\). The second policy, Eq. (18), states that when the existing inventory level exceeds the optimal level, we do not order. All of which seems to make good common sense.

The particularly simple ordering policy discussed above has been the source of great interest and investigation. Others have investigated the conditions under which other simple ordering policies exist [2].

8. Numerical Example

Let us consider the following case
\[ \varphi(s) = e^{-s}, \quad p = 2 \]
\[ a = 0.9, \quad k = 1.5 \]
We see that these terms satisfy the various assumptions in Section 7. To evaluate the \( y = \bar{x} \), by Eq. (16) of Section 7.

\[
\int_0^y \varphi(s) \, ds = \frac{ap - k}{a(p - k)} = \frac{(0.9) (2) - 1.5}{0.9(2 - 1.5)} = 0.667
\]

\[
\int_0^y \varphi(s) \, ds = \int_0^y e^{-s} \, ds = -e^{-y} + 1 = 0.667
\]

\[ y = \bar{x} = 1.1 \]

9. Ordering Cost with Setup Charges

The above discussion for simple proportional costs and penalty yielded a particularly simple and “practical” type of solution. The crux of the solution is the fact that the minimizing equation [Eq. (9)] of Section 7 has only one root. The uniqueness of the root is responsible for the simple structure of the solution. Any cost function term that adds to the multiplicity of roots to the minimizing equation inherently complicates the solution. To illustrate the point let us consider the ordering cost to be \( p(s - y) + q \). Here is \( q \) is a constant setup or fixed charge or administrative cost, which is always present and is independent of the quantity actually ordered.

The counterpart to Eq. (8) of Section 7 is

\[
f(x) = \min_{y \geq x} \left[ k(y - x) + a \int_y^\infty (p(s - y) + q) \varphi(s) \, ds \right.
\]

\[
+ a f(0) \int_y^\infty \varphi(s) \, ds + a \int_0^y f(y - s) \varphi(s) \, ds \right]
\]

(1)

This expression differs from (8) of Section 7 only in the term \( aq \int_y^\infty \varphi(s) \, ds \).

On taking the derivative of the right-hand side and setting it equal to zero, we find

\[
0 = k + a \left[ -p \int_y^\infty \varphi(s) \, ds - q\varphi(y) + \int_y^\infty f'(y - s) \varphi(s) \, ds \right]
\]

(2)
Since
\[ f'(y - s) = \frac{-k}{y} \quad y > x \quad (3) \]
by Eq. (11) of Section 7, then Eq. (2) may be expressed as
\[
0 = k + a \left[ -p \int_y^\infty \varphi(s) \, ds - q\varphi(y) - k \int_0^y \varphi(s) \, ds \right] \quad (4)
\]
Comparing this equation to (12) of Section 7, we see the one additional term \( q\varphi(y) \). This equation does not possess in general a unique root for all density functions. The structure of the solution depends on the \( \varphi(s) \). If \( \varphi'(s) \leq 0 \), then it can be shown that (4) does have a unique root.

10. Discrete Finite Time Period Model

Up to this point we have considered only continuous processes where the ordering, penalties, and demands were continuous functions of time. In many practical situations discrete stage processes may represent more closely operating practice. For example, company policy may be to order only once a week.

We consider here that orders are placed at equally spaced intervals and are immediately filled. After the order has been made and filled, the demand is made and satisfied. If the demand can not be met, the penalty is invoked. A short enough time period is taken so that the discount factor is not required.

Let us define
\[
f_N(x) = \text{the expected cost over an } N\text{-stage period starting with an initial inventory } x \text{ and using an optimal policy}
\]
In a manner similar to the infinite stage process, we have
\[
f_1(x) = \min_{y \geq x} \left[ k(y - x) + p \int_y^\infty (s - y) \varphi(s) \, ds \right] \quad (1)
\]
\[
f_N(x) = \min_{y \geq x} \left[ k(y - x) + p \int_y^\infty (s - y) \varphi(s) \, ds + f_{N-1}(0) \int_y^\infty \varphi(s) \, ds \right. \\
\left. + \int_0^y f_{N-1}(y - s) \varphi(s) \, ds \right]
\]
It can be shown in a manner similar to that for the infinite stage case that for \( p \geq k \) the optimal ordering policy is

\[
\begin{align*}
    y &= x_N; \quad x \leq x_N \\
    y &= x_N; \quad x > x_N
\end{align*}
\]

The sequence \( x_N \) increases monotonically in \( N \). This policy structure is the same as for the continuous case.

### 11. Inventory Model with Stochastic Input

In some inventory problems, the input into the inventory system is the random variable while the output may be considered fairly well known. A typical example of this is hydroelectric storage systems where the input to the system is the river flow that varies with rain and snow, and the output is the electrical supply. Another example is the purchase of material on the open market at a favorable price and the utilization of the commodity internally within the company at a known consumption pattern. In both of these examples the water flow rate and the price pattern are unknown in the future and decisions must be made on the expected cost or value of the inventory.

We will consider a discrete process with only a single commodity.

Let

\[
\begin{align*}
    v_i &= \text{the amount of commodity in inventory at the start of the } i\text{th time interval, } i = 0, 1, 2, ..., N \\
    x_i &= \text{random variable; the quantity of material coming into inventory, at time } i = 0, 1, 2, ..., N \\
    s_i &= \text{decision variable. The quantity of material removed from inventory, at time } i = 0, 1, 2, ..., N
\end{align*}
\]

\( f(x_N, ..., x_i, ..., x_0) \) = the conditional probability density function of \((x_N, x_{N-1}, ..., x_i)\) given \((x_{i-1}, ..., x_0)\)

\( C_i(v_i, x_i, s_i) \) = the cost of operating in the \( i\)th time interval

\[
K_i = \sum_{j=1}^{N} C_i = \text{total cost from } i\text{th through } N\text{th interval}
\]

\( E_i = E(K_i) = \text{expected value of } K_i \text{ in the } i\text{th interval} \)

Time is counted so \( i = 0 \) refers to the beginning of the process and \( i = N \) refers to the end.

By the definition of expected value, we have

\[
E(K_i) = \int_{-\infty}^{\infty} \ldots \int_{-\infty}^{\infty} K_i f(x_N, ..., x_i | x_{i-1}, ..., x_0) \, dx_N \ldots dx_i
\]
By the definition of conditional probability, we may write
\[ f(x_N, ..., x_{i+1}, x_i | x_{i-1}, ..., x_0) \, dx_N \ldots \, dx_i \]
\[ = f(x_N, ..., x_{i+1} | x_i, x_{i-1}, ..., x_0) \, dx_N \ldots \, dx_{i+1} \, f(x_i | x_{i-1}, ..., x_0) \, dx_i \]  
(9)

By Eqs. (6)–(8) we have
\[ K_i = \sum_{j=i}^N C_j = C_i + \sum_{j=i+1}^N C_j = C_i + K_{i+1} \]  
(10)
and
\[ E \left( \sum_{j=1}^N C_j \right) = E(K_{i+1}) \]
\[ = E_{i+1} = \int_{-\infty}^{\infty} \ldots \int_{-\infty}^{\infty} K_{i+1} f(x_N, ..., x_{i+1} | x_i, ..., x_0) \, dx_N \ldots \, dx_{i+1} \]  
(11)

Substituting (10) and (11) into (8) and using (9), we find
\[ E_i = \int_{-\infty}^{\infty} (C_i + E_{i+1}) f(x_i | x_{i-1}, ..., x_0) \, dx_i \]  
(12)
where we note that \( C_i \) is not a function of the \( x_i \) after the \( i \)th interval.

Equation (12) gives a relationship between \( E_i \) and \( E_{i+1} \). If a starting place can be found where \( E_{i+1} \) is known, then the entire sequence of \( E_i \) terms can be developed. In this process we consider that at the end of the process \( E_N = 0 \), so this gives us a starting place.

Let us assume that the random variable density function for successive intervals is a simple Markov process
\[ f(x_i | x_{i-1}, ..., x_0) = f(x_i | x_{i-1}) \]  
(13)
That is to say, the conditional probability depends only on the value of the random variable one time unit earlier.

We assume here that \( s_i \) the quantity removed from inventory during the \( i \)th time period is a function of \( v_i \) and \( x_{i-1} \)
\[ s_i = s_i(v_i, x_{i-1}) \]  
(14)

\[ \text{\textsuperscript{*}} \text{The reader can satisfy himself by expanding both sides of Eq. (9)} \]
\[ f(x_N, ..., x_{i+1}, x_i | x_{i-1}, ..., x_0) = \frac{f(x_N, ..., x_{i+1}, x_i, x_{i-1}, ..., x_0)}{f(x_{i-1}, ..., x_0)} \]
\[ f(x_N, ..., x_{i+1} | x_i, x_{i-1}, ..., x_0) \cdot f(x_i | x_{i-1}, ..., x_0) \]
\[ = \frac{f(x_N, ..., x_{i+1}, x_i, x_{i-1}, ..., x_0)}{f(x_{i-1}, ..., x_0)} \cdot \frac{f(x_i | x_{i-1}, ..., x_0)}{f(x_{i-1}, ..., x_0)} \]
\[ = \frac{f(x_N, ..., x_{i+1}, x_i, x_{i-1}, ..., x_0)}{f(x_{i-1}, ..., x_0)} \]
\[ f(x_N, ..., x_{i+1}, x_i | x_{i-1}, ..., x_0) = f(x_N, ..., x_{i+1} | x_i, x_{i-1}, ..., x_0) \cdot f(x_i | x_{i-1}, ..., x_0) \]
The inventory at time \((i + 1)\) is a function of the inventory, the input to the inventory, and the output from inventory one time unit earlier:

\[
v_{i+1} = v_{i+1}(v_i, x_i, s_i)
\]

The cost of operating at time \(i\) depends on \(v_i, x_i, s_i\):

\[
C_i = C_i(v_i, x_i, s_i)
\]

We desire to minimize \(E_i(v_i, x_{i-1}, s_i)\) by manipulating \(s_i\). The functional equations are

\[
E_i(v_i, x_{i-1}) = \text{Min} \int_{-\infty}^{\infty} [C_i(v_i, x_i, s_i) + E_{i-1}(v_{i-1}, x_i)] f(x_i | x_{i-1}) \, dx_i
\]

and

\[
E_N = 0
\]

In (17) and (18) we observe once again the transliteration of the Principle of Optimality; Eqs. (17) and (18) not only describe the inventory process but also give a working computing algorithm.

We may compare this problem with the problem described in Sections 6 and 7. In this problem the amount of material coming into inventory \(x_i\) is the random variable. Its density function is known and is given by the joint probability \(f(x_N, ..., x_i | x_{i-1}, ..., x_0)\). In the problem in Section 6, the random variable is the demand upon the system inventory. In this problem the inventory level and cost are controlled by the quantity of material shipped from inventory. In this problem the inventory level and cost are controlled by the quantity of material coming into inventory. In the problem in Section 6 the cost was controlled by regulating the quantity of material coming into inventory. In this section the costs are defined only in terms of the current inventory level, the current input material to inventory, and the current material removed from storage. In the previous problem the cost function was specifically spelled out as the cost of ordering and the penalty cost for failure to meet the demands.

The two principal differences between the two inventory models is (1) that in one case the stochastic aspects appear on the inflow to the inventory, while in the other case the stochastic aspects appear in the demand pattern; (2) the probability density function is specified as a conditional probability function of a simple Markovian properties, while the other does not specify the nature of the density function.
12. Purchasing Materials on Fluctuating Market

An important inventory problem is the proper timing of purchases on a fluctuating market. The idea is, of course, to buy low and sell high, but the problem is to know what is low and what is high. It is necessary in this development to have a predictor or forecast for the prices in order to make rational decisions.

The following assumptions are made

1. Price and usage of the commodity are random variables in any time period.
2. The price and usage are independent of each other during any given time period and in the subsequent time periods.
3. A commodity once purchased, must be used, it can not be resold.
4. If the amount on hand in inventory is short of the amount needed, the difference is purchased in the open market at fixed premium cost.
5. There are no time lags, that is, the delivery of the commodity is instantaneous.
6. Inventory costs and the cost of shortages are known.

We define the following terms:

- $s$ = the current inventory on hand
- $S$ = the new level of inventory after a purchase
- $P_i$ = the current market price of the commodity at period $i$
- $N$ = the number of time periods
- $f_N(s, P_N)$ = the expected minimum purchasing, holding, and shortage costs over the $N$ time periods beginning in state $(s, P_N)$ and using an optimal policy
- $p$ = the future purchase prices of the commodity in the market
- $q(p)$ = the probability density function of the future price $p$
- $r$ = the usage of the stock
- $I(S)$ = the expected holding and shortage costs for the new inventory level, $S$

Using the Principle of Optimality we write

$$f_N(s, P_N) = \min_{S \geq s} \left[ (S - s) P_N + I(S) + \int_0^{\infty} \int_0^S f_{N-1}(S - r, p) \Phi(r) q(p) \, dr \, dp \right.$$ 

$$+ \int_0^{\infty} \int_S^{\infty} f_{N-1}(0, p) \Phi(r) q(p) \, dr \, dp \right]$$

(1)

$$f(s_1, P_1) = \min_{S \geq s} [(S - s) P_1 + I(S)]$$

(2)
The term \((S - s)\) represents the size of the order. The term \((S - s)P_N\) represents the cost of increasing the inventory from \(s\) to \(S\) at a price of \(P_N\) dollars per unit. In Eq. (1) the first two terms represent the cost of purchasing \((S - s)\) units and the holding cost for \(S\) units. The third term in (1) is the minimum expected cost over the \((N - 1)\) remaining stages when the new inventory exceeds the usage of the stock. The fourth term is similar to the third term except the inventory level is zero. In Eq. (1), the expectation is taken over two density functions, the price and the usage. Equation (2) represents the one-stage process.

Now an expression can be developed for \(I(S)\) as follows, referring to Fig. 3.

Over one period the average holding cost at usage rate \(r_1\) is

\[
C_1 = c_1(S - \frac{1}{2}r_1)
\]  

(3)

where \(c_1\) is the unit holding cost

For usage \(r_2\), where the stock is depleted in time \(t\), where \(t < T\), the average holding cost is

\[
C_2 = \frac{1}{2} c_1 S(t/T)
\]  

(4)

but

\[
\frac{t}{T} = \frac{S}{r_2}
\]  

(5)

\[
C_2 = \frac{1}{2} c_1 (S^2/r_2)
\]  

(6)
The total shortage cost for \( r \geq S \) is

\[
C = c_2(r - s)
\]  

(7)

where \( c_2 \) is the unit shortage cost and \( (r - s) \) is the shortage.

The expected inventory cost is the sum of the expected holding cost plus the shortage costs. If \( \Phi(r) \, dr \) is the probability of the usage rate being between \( r \) and \( r + dr \), the expected inventory cost is

\[
I(S) = \int_{0}^{S} c_1(S - \frac{1}{2} r) \Phi(r) \, dr + \int_{S}^{\infty} \left[ \frac{c_1 S^2}{2r} + c_2 r - S \right] \Phi(r) \, dr
\]  

(8)

The first term represents the holding cost for the average stock when there is no shortage. The second term represents the sum of the holding cost for average stock when there is a shortage plus the cost of the shortage.

If we assume that the minimum occurs within the constraints \( S_{max} \geq S \geq r \), then we may differentiate the one-stage model in Eq. (2). The minimum is found at

\[
P_1 = -\frac{dI(S)}{dS}
\]  

(9)

Talking the derivative of (8) with respect to \( S \) we have

\[
\frac{dI}{dS} = c_1 \int_{0}^{S} \Phi(r) \, dr + c_1 S \int_{S}^{\infty} \frac{\Phi(r)}{r} \, dr - c_2 \int_{S}^{\infty} \Phi(r) \, dr
\]  

(10)

Substituting (10) in (9) we have

\[
P_1 = -c_1 \int_{0}^{S} \Phi(r) \, dr - c_1 S \int_{S}^{\infty} \frac{\Phi(r)}{r} \, dr + c_2 \int_{S}^{\infty} \Phi(r) \, dr
\]  

(11)

Equation (11) gives the relationship between the cost of purchasing for the one-stage process and the size of the new inventory for minimum cost.

The substitution of the value of \( S \) corresponding to \( P_1 \) in (11) into Eq. (2) determines the minimum cost for the one-stage process.

Equation (1) may be differentiated with respect to \( S \) and set equal to zero:

\[
P_N + \frac{dI(S)}{dS} + \int_{0}^{\infty} \int_{0}^{S} \frac{\partial f_{N-1}(S - r, p)}{\partial S} \Phi(r) \varphi(p) \, dr \, dp = 0
\]  

(12)
If (12) has a unique root, then this value of $S$ and its corresponding $P_N(s)$ can be substituted into (1) to yield the least cost over the $N$ stages.

Equations (9)–(12) together with (1) and (2) can be used to yield the minimizing sequence.

13. Production Control and Inventory Control

In the next few sections we discuss certain aspects of a production and inventory control model which lead to equations similar to those in Sections 6–10. We will formulate models for the minimization of production and inventory costs, the maximization of profit, and the maximum rate of return on an investment.

We assume that the demand pattern for the production is given by a normal distribution

$$q(s) = \frac{1}{\sigma \sqrt{2\pi}} \exp \left[ -\frac{1}{2} \left( \frac{s - E}{\sigma} \right)^2 \right]$$

where $s$ is the demand, $E$ is the average predicted demand, and $\sigma$ is the standard deviation. We also assume that there is a probability $\alpha$ of satisfying the demand $x$:

$$\int_{-\infty}^{\infty} q(s) \, ds \geq \alpha, \quad 0 \leq \alpha \leq 1$$

If we let

$$\theta(t) = \int_{0}^{t} \frac{1}{\sqrt{2\pi}} \exp \left( -\frac{1}{2} t^2 \right) \, dt$$

and let

$$t = \frac{z - E}{\sigma}$$

then Eq. (2) may be rewritten as

$$\theta \left( \frac{z - E}{\sigma} \right) - \theta(-\infty) \geq \alpha$$

or as

$$\theta \left( \frac{z - E}{\sigma} \right) \geq \alpha - 0.5$$

The transformations resulting in Eqs. (5) and (6) will prove useful, as shown below.
The production and inventory model we construct consists of three costs: the cost of production, the cost of carrying inventory for over production, and the penalty cost for under production.

We define

\[ f_N(x) = \text{the expected cost of an } N\text{-time stage process beginning with inventory } x \text{ and using an optimal policy} \]

\[ r(y) = \text{the cost function of producing } y \text{ items} \]

\[ q(y) = \text{the cost function of carrying } y \text{ items in inventory (over production)} \]

\[ c(y) = \text{the penalty cost function for under production} \]

\[ \varphi(s) = \text{the demand distribution} \]

\[ z = \text{terminal inventory level at the end of a period} \]

\[ \delta = \text{discount factor (} \delta < 1 \text{)} \]

The one-stage process is described by

\[ f_1(x) = \min_{z \geq x} \left[ r(z - x) + \int_{-\infty}^{z} q(z - s) \varphi(s) \, ds + \int_{z}^{\infty} c(s - z) \varphi(s) \, ds \right] \tag{7} \]

while the \( N \)-stage process is described by

\[ f_N(x) = \min_{z \geq x} \left[ r(z - x) + \int_{-\infty}^{z} q(z - s) \varphi(s) \, ds \right. \]

\[ \left. + \int_{z}^{\infty} c(s - z) \varphi(s) \, ds + \delta \int_{-\infty}^{\infty} f_{N-1}(z - s) \varphi(s) \, ds \right] \tag{8} \]

The first term in Eq. (8) is the cost of production, the second term is the expected cost of carrying the over production, the third term is the expected penalty cost for under production, the last term is the discounted expected cost over the \((N - 1)\) remaining stages of the process. We take \( z \geq x \) since we do not manufacture negative production. The argument in \( f_{N-1}(z - s) \) may take on negative values indicating negative inventory.

If we specify linear cost functions as follows:

\[ r(t) = kt + \zeta \tag{9} \]

\[ q(t) = mt + \eta \tag{10} \]

\[ c(t) = \lambda t + \epsilon \tag{11} \]

and if we let \( \varphi(s) \) be the normal distribution, we can find analytical solutions to Eqs. (7) and (8).
Equations (7) and (8) now appear as

\[
\begin{align*}
\text{Equation } (7) & = \text{Min}_{z \geq x} \left[ k(z - x) + \zeta + \int_{-\infty}^{z} [\mu(z - s) + \eta] \varphi(s) \, ds \right. \\
& \quad \quad + \left. \int_{z}^{\infty} [\lambda(s - z) + \epsilon] \varphi(s) \, ds \right] \\
\text{Equation } (8) & = \text{Min}_{z \geq x} \left[ k(z - x) + \zeta + \int_{-\infty}^{z} [\mu(z - s) + \eta] \varphi(s) \, ds \right. \\
& \quad \quad + \left. \int_{z}^{\infty} [\lambda(s - z) + \epsilon] \varphi(s) \, ds \right]
\end{align*}
\]

Equation (13) may be solved by taking the derivative with respect to \( z \) of the right-hand side and setting it equal to zero. (This assumes that the minimum does exist.) This yields

\[
\begin{align*}
& k + \mu \int_{-\infty}^{z} \varphi(s) \, ds + \eta \varphi(z) - \lambda \int_{z}^{\infty} \varphi(s) \, ds - \epsilon \varphi(z) \\
& \quad + \delta \int_{-\infty}^{\infty} f_{N-1}(z - s) \varphi(s) \, ds = 0
\end{align*}
\]

By virtue of Eq. (10) of Section 7 and by virtue of \( \int_{-\infty}^{\infty} \varphi(s) \, ds = 1 \), the last term in Eq. (14) reduces to

\[
\delta \int_{-\infty}^{\infty} -k \varphi(s) \, ds = -\delta k
\]

Equation (14) may be written as

\[
(\eta - \epsilon) \varphi(z) - (\mu + \lambda) \int_{z}^{\infty} \varphi(s) \, ds = -\mu - k(1 - \delta)
\]  

In order for this equation to have a solution, it is necessary that the left-hand side of the equation be negative since the right-hand side is a negative number. If a solution exists, uniqueness is guaranteed by \( \varphi'(s) \leq 0 \).

Since the constants \( \eta, \epsilon, \mu, \lambda, k, \delta \) are all known, the optimal value of \( z \) may be found by a trial and error solution of (15) for various values of \( z \). Zachary, on the other hand, employs a more elegant solution, which we describe below [20].
We consider two cases, \( \eta - \epsilon = 0 \) and \( \eta - \epsilon \neq 0 \). For \( \eta - \epsilon = 0 \), Eq. (15) reduces to

\[
\int_{z}^{\infty} \varphi(s) \, ds = \frac{\mu + k(1 - \delta)}{\mu + \lambda} \quad (16)
\]

or

\[
\theta \left( \frac{z - E}{\sigma} \right) = \left[ 0.5 - \frac{k(1 - \delta) + \mu}{\mu + \lambda} \right] \quad (17)
\]

Since the left-hand side of (16) is a fraction by definition of a probability density function, the following conditions must hold for an \( N \)-stage process and for a one-stage process:

\[
0 < k(1 - \delta) < \lambda, \quad N \geq 2 \quad (18)
\]

\[
0 < k < \lambda, \quad N = 1 \quad (19)
\]

For \( \eta - \epsilon \neq 0 \), Eq. (15) may be transformed to a linear first-order ordinary differential equation by substituting

\[
\psi(z) = \int_{E}^{z} \varphi(s) \, ds \quad (20)
\]

Let us observe that

\[
\int_{-\infty}^{E} \varphi(s) \, ds + \int_{E}^{\infty} \varphi(s) \, ds + \int_{z}^{\infty} \varphi(s) \, ds = 1 \quad (21)
\]

and

\[
\int_{-\infty}^{E} \varphi(s) \, ds = 0.5 \quad (22)
\]

since \( E \) is the mean. The substitution of (20)–(22) into (15) gives

\[
\psi'(z) + \frac{\mu + \lambda}{\eta - \epsilon} \psi(z) = \frac{0.5(\lambda - \mu) - k(1 - \delta)}{\eta - \epsilon} \quad (23)
\]

The solution to this first-order linear ordinary differential equation, using the boundary condition \( \psi(E) = 0 \), is

\[
\psi(z) = \left[ 0.5 - \frac{k(1 - \delta) + \mu}{\lambda + \mu} \right] \left[ 1 - \exp \left( - \left( \frac{\lambda + \mu}{\eta - \epsilon} \right) (z - E) \right) \right] \quad (24)
\]

Letting

\[
\alpha = 0.5 - \frac{k(1 - \delta) + \mu}{\lambda + \mu} \quad (25)
\]

\[
\beta = \frac{\lambda + \mu}{\eta - \epsilon} \quad (26)
\]
we may rewrite (24) as

$$\psi(z) = \alpha \left[ 1 - \exp \left[ - \beta \left( \frac{z - E}{\sigma} \right) \right] \right]$$  \hspace{1cm} (27)

Letting

$$z = E + \sigma y$$  \hspace{1cm} (28)

and using the definition of $\theta(t)$, in Eq. (2) we transform (27) into

$$\theta(y) = \int_0^y \frac{1}{\sqrt{2\pi}} \exp \left( - \frac{1}{2} y^2 \right) dy = \alpha(1 - e^{-\beta y})$$  \hspace{1cm} (29)

Zachary has tabulated the values of $y$ corresponding to a range of $\alpha$ and $\beta$ values, so a trial and error solution can be avoided by using his results (see Table 3). The significance of (29) is that it yields the inventory level as function of the average demand $E$, the standard deviation $\sigma$, and the parameters of the process embodied in $\alpha$ and $\beta$.

Since the $\theta(y)$ is a positive number, the signs of $\alpha$ and $\beta$ must be such to make the right-hand side also positive. In particular, if $\alpha$ and $\beta$ are of like signs, positivity is guaranteed. These comments about the signs of $\alpha$ and $\beta$ are equivalent to the remarks made regarding the necessary condition for solving (15).

14. Maximum Profit for Production and Inventory Control

We augment the production and inventory control model discussed in Section 13 by including the profit accrued by shipping the products. The expected value of the products shipped is given by

$$v = p \left[ \int_{-\infty}^{z} s \varphi(s) \, ds + x \int_{-\infty}^{z} \varphi(s) \, ds \right]$$  \hspace{1cm} (1)

where $p$ is the unit value of the product.

The first term in the brackets is the expected quantity of product shipped when the demand does not exceed inventory. The second term is the expected shipment when demand exceeds inventory. In this case the entire inventory is available for shipment.

The expected maximum profit is found from the algebraic sum of the
profit from shipping less the costs of production, inventory holding, and penalty for under production. The functional equations are

\[ g_1(x) = \max_{z \geq x} \left[ p \int_{-\infty}^{z} sp(s) \, ds + pz \int_{z}^{\infty} \varphi(s) \, ds - k(z - x) - \xi \right] \]

\[ - \int_{-\infty}^{\infty} [\mu(z - s) + \eta] \varphi(s) \, ds - \int_{z}^{\infty} [\lambda(s - z) + \epsilon] \varphi(s) \, ds \]  

\[ g_N(x) = \max_{z \geq x} \left[ p \int_{-\infty}^{z} sp(s) \, ds + pz \int_{z}^{\infty} \varphi(s) \, ds - k(z - x) - \xi \right] \]

\[ - \int_{-\infty}^{\infty} [\mu(z - s) + \eta] \varphi(s) \, ds - \int_{z}^{\infty} [\lambda(s - z) + \epsilon] \varphi(s) \, ds \]

\[ + \delta \int_{-\infty}^{\infty} g_{N-1}(z - s) \varphi(s) \, ds \]  

The first two terms in these expressions come from Eq. (1), while the last three terms are taken from Eqs. (12) and (13) of Section 13, with the exception that \( g_{N-1}(z - s) \) replaces \( f_{N-1}(z - s) \).

The maximum of the right-hand side of (3), if it exists, is found by taking the derivative with respect to \( z \) and setting it equal to zero. The resulting expression is

\[ p \int_{z}^{\infty} \varphi(s) \, ds - k - \mu \int_{-\infty}^{z} \varphi(s) \, ds - \eta \varphi(z) + \lambda \int_{z}^{\infty} \varphi(s) \, ds + \epsilon \varphi(z) + \delta k = 0 \]  

Using Eqs. (21) and (22) of Section 13, this simplifies to

\[ (\eta - \epsilon) \varphi(z) + (p + \mu + \lambda) \int_{E}^{z} \varphi(s) \, ds = 0.5(p + \mu + \lambda) - k(1 - \delta) - \mu \]  

If \( \eta - \epsilon = 0 \), we have

\[ \int_{E}^{z} \varphi(s) \, ds = 0.5 - \frac{k(1 - \delta) + \mu}{p + \lambda + \mu} \]  

The restriction on the parameters is

\[ 0 \leq 0.5 - \frac{k(1 - \delta) + \mu}{p + \lambda + \mu} \leq 1 \]  

If \( \eta - \epsilon \neq 0 \), we may take as before \( \psi(z) = \int_{E}^{z} \varphi(s) \, ds \) and substitute \( \psi(z) \) into (5). Equation (5) is transformed into a first-order ordinary differential equation

\[ (\eta - \epsilon) \psi'(z) + (p + \mu + \lambda) \psi(z) = 0.5(p + \mu + \lambda) - k(1 - \delta) - \mu \]
TABLE 3
\[
\int_0^\beta \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2} y^2\right) dy = \alpha(1 - e^{-\beta})
\]

<table>
<thead>
<tr>
<th>( \beta )</th>
<th>10</th>
<th>20</th>
<th>30</th>
<th>40</th>
<th>50</th>
<th>60</th>
<th>70</th>
<th>80</th>
<th>90</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.001</td>
<td>-0.5316</td>
<td>-0.2250</td>
<td>-0.1329</td>
<td>-0.09025</td>
<td>-0.06622</td>
<td>-0.05102</td>
<td>-0.04066</td>
<td>-0.03321</td>
<td>-0.02764</td>
</tr>
<tr>
<td>0.002</td>
<td>-0.4469</td>
<td>-0.1802</td>
<td>-0.1020</td>
<td>-0.06640</td>
<td>-0.04666</td>
<td>-0.03435</td>
<td>-0.02607</td>
<td>-0.02019</td>
<td>-0.01584</td>
</tr>
<tr>
<td>0.003</td>
<td>-0.3956</td>
<td>-0.1528</td>
<td>-0.08286</td>
<td>-0.05151</td>
<td>-0.03434</td>
<td>-0.02376</td>
<td>-0.01673</td>
<td>-0.01180</td>
<td>-0.008187</td>
</tr>
<tr>
<td>0.004</td>
<td>-0.3383</td>
<td>-0.1326</td>
<td>-0.06866</td>
<td>-0.04037</td>
<td>-0.02505</td>
<td>-0.01573</td>
<td>-0.009595</td>
<td>-0.005336</td>
<td>-0.002255</td>
</tr>
<tr>
<td>0.005</td>
<td>-0.3287</td>
<td>-0.1165</td>
<td>-0.05721</td>
<td>-0.03131</td>
<td>-0.01745</td>
<td>-0.009106</td>
<td>-0.00368</td>
<td>0.000</td>
<td>0.002721</td>
</tr>
<tr>
<td>0.006</td>
<td>-0.3040</td>
<td>-0.1029</td>
<td>-0.04751</td>
<td>-0.02359</td>
<td>-0.01093</td>
<td>-0.003384</td>
<td>0.001468</td>
<td>0.004762</td>
<td>0.007091</td>
</tr>
<tr>
<td>0.007</td>
<td>-0.2827</td>
<td>-0.09113</td>
<td>-0.03903</td>
<td>-0.01679</td>
<td>-0.005146</td>
<td>0.001714</td>
<td>0.006073</td>
<td>0.009003</td>
<td>0.01105</td>
</tr>
<tr>
<td>0.008</td>
<td>-0.2640</td>
<td>-0.08077</td>
<td>-0.03145</td>
<td>-0.01067</td>
<td>0.000</td>
<td>0.006349</td>
<td>0.01029</td>
<td>0.01471</td>
<td></td>
</tr>
<tr>
<td>0.009</td>
<td>-0.2472</td>
<td>-0.07122</td>
<td>-0.02456</td>
<td>-0.00577</td>
<td>0.004897</td>
<td>0.01063</td>
<td>0.01421</td>
<td>0.01655</td>
<td>0.1814</td>
</tr>
<tr>
<td>0.010</td>
<td>-0.2320</td>
<td>-0.06258</td>
<td>-0.01821</td>
<td>0.000</td>
<td>0.009376</td>
<td>0.01465</td>
<td>0.01790</td>
<td>0.01999</td>
<td>0.02141</td>
</tr>
<tr>
<td>0.020</td>
<td>-0.1248</td>
<td>0.0003012</td>
<td>0.02932</td>
<td>0.04001</td>
<td>0.04479</td>
<td>0.04717</td>
<td>0.04844</td>
<td>0.04915</td>
<td>0.04956</td>
</tr>
<tr>
<td>0.030</td>
<td>-0.05452</td>
<td>0.04401</td>
<td>0.06429</td>
<td>0.07081</td>
<td>0.07331</td>
<td>0.07438</td>
<td>0.07484</td>
<td>0.07506</td>
<td>0.07516</td>
</tr>
<tr>
<td>0.040</td>
<td>0.0004844</td>
<td>0.08014</td>
<td>0.09449</td>
<td>0.09844</td>
<td>0.09972</td>
<td>0.1002</td>
<td>0.1003</td>
<td>0.1004</td>
<td>0.1004</td>
</tr>
<tr>
<td>0.050</td>
<td>0.04698</td>
<td>0.1123</td>
<td>0.1224</td>
<td>0.1248</td>
<td>0.1254</td>
<td>0.1256</td>
<td>0.1256</td>
<td>0.1256</td>
<td>0.1256</td>
</tr>
<tr>
<td>0.060</td>
<td>0.08827</td>
<td>0.1421</td>
<td>0.1492</td>
<td>0.1506</td>
<td>0.1509</td>
<td>0.1509</td>
<td>0.1509</td>
<td>0.1509</td>
<td>0.1509</td>
</tr>
<tr>
<td>0.070</td>
<td>0.1260</td>
<td>0.1705</td>
<td>0.1754</td>
<td>0.1762</td>
<td>0.1763</td>
<td>0.1763</td>
<td>0.1764</td>
<td>0.1764</td>
<td>0.1764</td>
</tr>
<tr>
<td>0.080</td>
<td>0.1612</td>
<td>0.1980</td>
<td>0.2014</td>
<td>0.2018</td>
<td>0.2019</td>
<td>0.2019</td>
<td>0.2019</td>
<td>0.2019</td>
<td>0.2019</td>
</tr>
<tr>
<td>0.090</td>
<td>0.1946</td>
<td>0.2250</td>
<td>0.2273</td>
<td>0.2275</td>
<td>0.2275</td>
<td>0.2275</td>
<td>0.2275</td>
<td>0.2275</td>
<td>0.2275</td>
</tr>
<tr>
<td>0.100</td>
<td>0.2266</td>
<td>0.2517</td>
<td>0.2532</td>
<td>0.2533</td>
<td>0.2533</td>
<td>0.2533</td>
<td>0.2533</td>
<td>0.2533</td>
<td>0.2533</td>
</tr>
<tr>
<td>0.150</td>
<td>0.3759</td>
<td>0.3852</td>
<td>0.3853</td>
<td>0.3853</td>
<td>0.3853</td>
<td>0.3853</td>
<td>0.3853</td>
<td>0.3853</td>
<td>0.3853</td>
</tr>
<tr>
<td>0.200</td>
<td>0.5213</td>
<td>0.5244</td>
<td>0.5244</td>
<td>0.5244</td>
<td>0.5244</td>
<td>0.5244</td>
<td>0.5244</td>
<td>0.5244</td>
<td>0.5244</td>
</tr>
<tr>
<td>0.250</td>
<td>0.6736</td>
<td>0.6745</td>
<td>0.6745</td>
<td>0.6745</td>
<td>0.6745</td>
<td>0.6745</td>
<td>0.6745</td>
<td>0.6745</td>
<td>0.6745</td>
</tr>
<tr>
<td>0.300</td>
<td>0.8414</td>
<td>0.8416</td>
<td>0.8616</td>
<td>0.8616</td>
<td>0.8616</td>
<td>0.8616</td>
<td>0.8616</td>
<td>0.8616</td>
<td>0.8616</td>
</tr>
<tr>
<td>0.350</td>
<td>1.036</td>
<td>1.036</td>
<td>1.036</td>
<td>1.036</td>
<td>1.036</td>
<td>1.036</td>
<td>1.036</td>
<td>1.036</td>
<td>1.036</td>
</tr>
<tr>
<td>0.400</td>
<td>1.281</td>
<td>1.281</td>
<td>1.281</td>
<td>1.281</td>
<td>1.281</td>
<td>1.281</td>
<td>1.281</td>
<td>1.281</td>
<td>1.281</td>
</tr>
<tr>
<td>0.450</td>
<td>1.645</td>
<td>1.645</td>
<td>1.645</td>
<td>1.645</td>
<td>1.645</td>
<td>1.645</td>
<td>1.645</td>
<td>1.645</td>
<td>1.645</td>
</tr>
</tbody>
</table>
Using the boundary condition \( \psi(E) = 0 \), we find

\[
\psi(x) = \frac{0.5(p - \mu + \lambda) - k(1 - \delta) - \mu}{\rho + \mu + \lambda} \left[ 1 - \exp \left[ - \left( \frac{p + \lambda + \mu}{\eta - \epsilon} \right) (x - E) \right] \right] \quad (9)
\]

If we let

\[
\alpha = \frac{k(1 - \delta) + \mu}{\rho + \mu + \lambda} \quad (10)
\]
and

\[
\beta = \left( \frac{p + \mu + \lambda}{\eta - \epsilon} \right) \sigma \quad (11)
\]
we have

\[
\psi(x) = \alpha \left[ 1 - \exp \left[ - \beta \left( \frac{x - E}{\sigma} \right) \right] \right] \quad (12)
\]

Letting

\[
z = E + \sigma y \quad (13)
\]
and using the definition of \( \theta(t) \) from Section 13, we write (12) as

\[
\theta(y) = \int_{0}^{y} \frac{1}{\sqrt{2\pi}} \exp \left( - \frac{1}{2} y^{2} \right) dy = \alpha (1 - e^{-\theta y}) \quad (14)
\]

This equation is identical in form to Eq. (29) of Section 13 except the \( \alpha \) and \( \beta \) terms are defined by (10) and (11). As a consequence of this similarity, the tabled \( \alpha, \beta, y \) in Table 3 still hold.

### 15. Return on Investment

An interesting aspect of the problem discussed in Section 14 is the optimizing of the rate of return. Let us define the rate of return at the ratio of the profit per stage divided by the investment per stage. The functional expression for maximizing the rate of return is

\[
R_{N}(x) = \max \left[ p \int_{-\infty}^{x} s \varphi(s) \ ds + px \int_{x}^{\infty} q(s) \ ds - k(x - x) - \zeta \right.
\]

\[
- \int_{-\infty}^{x} \left[ \mu(x - s) + \eta \right] \varphi(s) \ ds - \int_{x}^{\infty} \left[ \lambda(s - z) + \epsilon \right] \varphi(s) \ ds \bigg]\bigg/ C_{N}
\]

\[
+ \delta \int_{-\infty}^{\infty} R_{N-1}(z - s) \varphi(s) \ ds \quad (1)
\]

where \( C_{N} \) = the investment during period \( N \).
The maximum of the bracketed term, if it exists, is given by

$$p \int_{x}^{\infty} \varphi(s) \, ds - \mu \int_{-\infty}^{x} \varphi(s) \, ds + \lambda \int_{x}^{\infty} \varphi(s) \, ds + (\epsilon - \eta) \varphi(x) + \left( \frac{C_{N} \delta}{C_{N-1}} - 1 \right) k = 0$$  \hspace{1cm} (2)

If we redefine $\psi(z)$ as

$$\psi(z) = \int_{-x}^{x} \varphi(s) \, ds$$  \hspace{1cm} (3)

and substitute it into (2), we obtain

$$(\eta - \epsilon) \psi'(z) + (p + \lambda + \mu) \psi(z) = p + \lambda + k \left( \frac{C_{N} \delta}{C_{N-1}} - 1 \right)$$  \hspace{1cm} (4)

If $\eta - \epsilon = 0$, we write (4) as

$$\psi(z) = \frac{p + \lambda + k \left( \frac{C_{N} \delta}{C_{N-1}} - 1 \right)}{p + \lambda + \mu}$$  \hspace{1cm} (5)

If $\eta - \epsilon \neq 0$, the solution of (4), using the boundary condition $\psi(E) = 0.5$, is given by

$$\psi(z) = \frac{p + \lambda + k \left( \frac{C_{0} \delta}{C_{1}} - 1 \right)}{p + \lambda + \mu} + \left[ 0.5 - \frac{p + \lambda + k \left( \frac{C_{0} \delta}{C_{1}} - 1 \right)}{p + \lambda + \mu} \right] \times \exp \left[ - \left( \frac{p + \lambda + \mu}{\eta - \epsilon} \right) (z - E) \right]$$  \hspace{1cm} (6)

where $C_{0} = C_{N}$ and $C_{N-1} = C_{1}$.

Recalling that

$$\psi(z) = \theta \left( \frac{z - E}{\sigma} \right) - \theta(-\infty) = \theta \left( \frac{z - E}{\sigma} \right) + 0.5$$  \hspace{1cm} (7)

we may express (6) as

$$\theta \left( \frac{z - E}{\sigma} \right) = -0.5 + \frac{p + \lambda + k \left( \frac{C_{0} \delta}{C_{1}} - 1 \right)}{p + \lambda + \mu} \left[ 0.5 - \frac{p + \lambda + k \left( \frac{C_{0} \delta}{C_{1}} - 1 \right)}{p + \lambda + \mu} \right] \exp \left[ - \frac{p + \lambda + \mu}{\eta - \epsilon} (z - E) \right]$$  \hspace{1cm} (8)
If we let

$$\alpha = -0.5 + \frac{p + \lambda + k \left( \frac{C \delta}{C_1} - 1 \right)}{p + \lambda + \mu}$$  \hspace{1cm} (9)$$

$$\beta = \frac{p + \lambda + \mu}{\eta - \epsilon} - \sigma$$  \hspace{1cm} (10)$$

$$y = \frac{z - E}{\sigma}$$  \hspace{1cm} (11)$$

then Eq. (8) becomes

$$\theta(y) = \alpha (1 - e^{-\beta y})$$  \hspace{1cm} (12)$$

Once again the tabled values of $\alpha$, $\beta$, $y$ are applicable.

16. In Plant Inventory Problem

In many chemical engineering applications the raw material for a process is stored as well as the products from the process. The products from the process may in turn be the feed material for other processes. The plant operator must operate the process so the raw material inventory never builds up too high or depletes too low. Similarly, the product inventory must be held in balance. Both the supply of raw material to the process and the product inventory may be considered deterministic or stochastic. In fact, four possible combinations of inventory conditions exist.

<table>
<thead>
<tr>
<th>Supply conditions</th>
<th>Product conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deterministic</td>
<td>Deterministic</td>
</tr>
<tr>
<td>Deterministic</td>
<td>Stochastic</td>
</tr>
<tr>
<td>Stochastic</td>
<td>Deterministic</td>
</tr>
<tr>
<td>Stochastic</td>
<td>Stochastic</td>
</tr>
</tbody>
</table>

If we consider one raw material and one product, the system is sketched in Fig. 4.

Let us suppose that it is desired to minimize the cost of operating the two inventories and the process itself. In order to evaluate the cost of the supply and product inventories, let us measure arbitrarily our discomfiture by the following scale. If the level in a tank is approximately one-half full, there is no storage cost. If the level is outside this region, we
assign higher costs. To be specific, for the supply tank, the cost-inventory structure is given as

\[
\begin{align*}
C &= 0, \quad h = \bar{h} \pm 2\Delta h \\
C &= C_1, \quad \bar{h} \pm 2\Delta h < h \leq \bar{h} \pm 4\Delta h \\
C &= C_2, \quad \bar{h} \pm 4\Delta h < h \leq \bar{h} \pm 6\Delta h \\
C &= C_3, \quad \bar{h} \pm 6\Delta h < h
\end{align*}
\]  

(1)

where \( C_3 > C_2 > C_1 > 0 \), and \( \bar{h} = \text{mid-point inventory level in the tank} \), and \( \Delta h \) is a fixed incremental volume

Similarly, for the product inventory, the cost-inventory structure is given as

\[
\begin{align*}
K &= 0, \quad H = \bar{H} \pm 2\Delta H \\
K &= K_1, \quad \bar{H} \pm 2\Delta H < H \leq \bar{H} \pm 4\Delta H \\
K &= K_2, \quad \bar{H} \pm 4\Delta H < H \leq \bar{H} \pm 6\Delta H \\
K &= K_3, \quad \bar{H} \pm 6\Delta H < H
\end{align*}
\]  

(2)

where \( K_3 > K_2 > K_1 > 0 \) and \( \bar{H} = \text{mid-point inventory level} \).

Let us suppose that there is a cost of operating the unit \( g(y) \) where \( y \) is the quantity of raw material processed in the unit.

The total cost of operating the system is therefore over \( N \) stages of time

\[
C = \sum_{i=1}^{N} [C_i(h) + g_i(y) + K_i(H)]
\]  

(3)
For the deterministic case for supply and demand, let

\[ y_i = \text{the quantity of raw material removed from the supply tank at time } i, \text{ to be processed by the reactor} \] (4)

\[ s_i = \text{the supply of raw material sent to the supply tank at time } i \] (known value) (5)

\[ p_i = \text{the demand for product as function of time at time } i \text{ (known value)} \] (6)

A material balance on the supply tank yields

\[ h_{i+1} = s_i + h_i - y_i \] (7)

Let \( m(y_i) \) be fraction of \( y_i \) that goes to product. Then a material balance on the product tank gives

\[ H_{i+1} = m(y_i) y_i + H_i - p_i \] (8)

A material balance over \( N \) time stages yields

\[ \sum_{i=1}^{N} s_i = \sum_{i=1}^{N} p_i + (h_1 - h_N) + (H_1 - H_N) \] (9)

Since the product demands can be met by either taking material out of storage and/or producing product, there is an optimization problem of how to choose the value of \( y_i \) over the \( N \) stages of time.

Let us define

\[ f_N(h, H) = \text{the minimum cost of supplying the product demand over } N \text{ stages subject to the supply inventory beginning in state } (h, H) \text{ and following an optimal policy} \] (10)

The minimization is expressed by

\[ f_N(h, H) = \min_{y_N} [C_N(h) + g_N(y_N) + K_N(H) + f_{N-1}(h', H')] \] (11)

where

\[ h' = \text{the new value of } h = h_{N-1} = s_N + h_N - y_N \] (12)

\[ H' = \text{the new value of } H = H_{N-1} = m(y_N) y_N + H_N - p_N \] (13)

\[ f_1(h, H) = \min_{y_1} [C_1(h) + g_1(y_1) + K_1(H)] \] (14)

The policy of \( y_i \) over the \( N \)-stage process depends on the nature of the supply and demand pattern, the relative cost of holding these inventories, and the cost of processing.
17. Smoothing Problem

Very closely related to inventory and scheduling problems are smoothing problems. These problems as pointed out in Chapter 4, Section 23 are characterized by production and/or inventory processes in which it is desired to modulate fluctuations in the manufacture and/or inventory of the products.

The smoothing problem that we desire to set up may be stated as follows:

Minimize the cost of over and under production and the cost of changing the rate of production when the process is subject to known fluctuations in the demand pattern over time.

Let

\[ r(t) = \text{the demand of product as a function of time} \]
\[ x(t) = \text{the production rate} \]
\[ F(x(t) - r(t)) = \text{cost of the net rate of production function} \]
\[ G(dx/dt) = \text{cost of change in the rate of production function} \]

Assuming that \( F \) and \( G \) are known functions, we require that the production always be equal or greater than the demand:

\[ x(t) \geq r(t) \quad (1) \]

The objective may be expressed as minimizing

\[ J(x) = \int_0^T \left[ F(x(t) - r(t)) + G \left( \frac{dx}{dt} \right) \right] dt \quad (2) \]

where \( x(t) \) is the manipulated variable.

For \( N\Delta = T \) and \( \Delta = 1 \) we may replace (2) by its discrete analog:

\[ J_1(x_k) = \sum_{k=1}^{N} F(x_k - r_k) + G(x_k - x_{k-1}) \quad (3) \]

We define

\[ f_M(c) = \min_{x_k} J_M(x_k) = \min_{x_k} \sum_{k=M}^{N} F(x_k - r_k) + G(x_k - x_{k-1}) \quad (4) \]

The initial condition is

\[ c = x_{M-1} \quad (5) \]

We count time forward so stage \( N \) is the last stage.
The functional equations describing the process, using the Principle of Optimality are

\[ f_M(c) = \min_{x_M^0} \left[ F(x_M - r_M) + G(x_M - c) + f_{M+1}(x_M) \right], \quad M = 1, 2, \ldots, N - 1 \]  

(6)

where \( c = x_{M-1} \);

\[ f_N(c) = \min_{x_N^0} \left[ F(x_N - r_N) + G(x_N - c) \right] \]  

(7)

where \( c = x_{N-1} \).

From Eqs. (6) and (7) the production policy can be generated to meet the demands upon the system.

Equations (6) and (7) are general equations for the smoothing process since neither the \( F \) or \( G \) functions have been specified. In the next section analytical results are developed using quadratic expressions for \( F \) and \( G \).

18. Quadratic Smoothing Problem

In Chapter 4, Section 23, we presented the smoothing problem as a calculus of variations problem for the continuous case using quadratic functions. In this section we will discuss the quadratic smoothing problem as a discrete problem. In particular, we will present some analytical results based on Section 17.

Our objective is to minimize over \( x_k \) the function

\[ J_1(x_k) = \sum_{k=1}^{N} b_k(x_k - d_k)^2 + e_k(x_k - x_{k-1})^2 \]  

(1)

where \( b_k, d_k, e_k \) are known numbers for \( k = 1, 2, \ldots, N \).

The first and second terms on the right in Eq. (1) correspond to the \( F \) and \( G \) terms respectively in Eq. (3) of Section 17.

We let

\[ x_{k-1} = x \]  

(2)

and we define†

\[ f_N(x) = \min_{x_N} \left[ b_N(x_N - d_N)^2 + e_N(x_N - x)^2 \right] \]  

(3)

where stage \( N \) refers to the last stage.

† Note: \( x_{k-1} = x \), as the initial condition corresponds to \( x_{k-1} = c \), as the initial condition in the Section 17.
Differentiating Eq. (3) with respect to $x_N$ and setting the expression equal to zero, we have

$$2b_N(x_N - d) + 2e_N(x_N - x) = 0$$  \hspace{1cm} (4) \\
$$(b_N + e_N)x_N - b_Nd_N - e_Nx = 0$$  \hspace{1cm} (5) \\
$$x_N = \frac{e_N}{b_N + e_N}x + \frac{b_Nd_N}{b_N + e_N}$$  \hspace{1cm} (6) \\
$$x_N = A_Nx + B_N$$  \hspace{1cm} (7)

where

$$A_N = \frac{e_N}{b_N + e_N}$$  \hspace{1cm} (8) \\
$$B_N = \frac{b_Nd_N}{b_N + e_N}$$  \hspace{1cm} (9)

Substituting (8) and (9) into (3) yields

$$f_N(x) = b_N(A_Nx + B_N - d_N)^2 + e_N(A_Nx + B_N - x)^2$$
$$= b_N[(A_Nx)^2 + 2A_N(B_N - d_N)x + (B_N - d_N)^2]$$
$$+ e_N[(A_N - 1)^2x^2 + 2(A_N - 1)B_Nx + B_N^2]$$

$$f_N(x) = [b_N(B_N - d_N)^2 + e_NB_N^2] + 2[b_NA_N(B_N - d_N) + e_N(A_N - 1)B_N]x$$
$$+ [b_NA_N^2 + e_N(A_N - 1)^2]x^2$$  \hspace{1cm} (10)

We may write Eq. (11) as

$$f_N(x) = u_N + v_Nx + w_Nx^2$$  \hspace{1cm} (12)

where

$$u_N = b_N(B_N - d_N)^2 + e_NB_N^2$$  \hspace{1cm} (13) \\
$$v_N = 2b_NA_N(B_N - d_N) + 2e_N(A_N - 1)B_N$$  \hspace{1cm} (14) \\
$$w_N = b_NA_N^2 + e_N(A_N - 1)^2$$  \hspace{1cm} (15)

Consider now the recursion relationship

$$f_{N-1}(x) = \text{Min}_{x_{N-1}}[b_{N-1}(x_{N-1} - d_{N-1})^2 + e_{N-1}(x_{N-1} - x)^2 + f_N(x_{N-1})]$$  \hspace{1cm} (16)
From (12) we write for \( f_N(x_{N-1}) \)

\[
f_N(x_{N-1}) = u_N + v_N x_{N_{-1}} + w_N x_{N-1}^2
\]  

(17)

Substituting (17) in (16) we have

\[
f_{N-1}(x) = \min_{x_{N-1}} \left[ b_{N-1}(x_{N-1} - d_{N-1})^2 + e_{N-1}(x_{N-1} - x)^2 + u_N + v_N x_{N_{-1}} + w_N x_{N-1}^2 \right]
\]  

(18)

Taking the derivative of Eq. (18) with respect to \( x_{N-1} \) and setting the expression equal to zero, we find

\[
2b_{N-1}(x_{N-1} - d_{N-1}) + 2e_{N-1}(x_{N-1} - x) + v_N + 2w_N x_{N-1} = 0
\]  

(19)

\[
x_{N-1}(2b_{N-1} + 2e_{N-1} + 2w_N) = 2e_{N-1}x + 2b_{N-1}d_{N-1} - v_N
\]  

(20)

\[
x_{N-1} = \frac{e_{N-1}x}{b_{N-1} + e_{N-1} + w_N} + \frac{b_{N-1}d_{N-1} - (v_N/2)}{b_{N-1} + e_{N-1} + w_N}
\]  

(21)

\[
x_{N-1} = A_{N-1}x + B_{N-1}
\]  

(22)

where

\[
A_{N-1} = \frac{e_{N-1}}{b_{N-1} + e_{N-1} + w_N}
\]  

(23)

\[
B_{N-1} = \frac{b_{N-1}d_{N-1} - (v_N/2)}{b_{N-1} + e_{N-1} + w_N}
\]  

(24)

Substituting (22), (23), and (24) into (18), we have

\[
f_{N-1}(x) = b_{N-1}[A_{N-1}x + B_{N-1} - d_{N-1})^2 + e_{N-1}(A_{N-1}x + B_{N-1} - x)^2
\]

\[
+ u_N + v_N(A_{N-1}x + B_{N-1})
\]

\[
+ w_N(A_{N-1}x^2 + 2A_{N-1}B_{N-1}x + B_{N-1}^2)
\]  

(25)

\[
f_{N-1}(x) = b_{N-1}[A_{N-1}x^2 + 2A_{N-1}(B_{N-1} - d_{N-1})x + (B_{N-1} - d_{N-1})^2]
\]

\[
+ e_{N-1}[(A_{N-1} - 1)^2 x^2 + 2(A_{N-1} - 1) B_{N-1}x + B_{N-1}^2]
\]

\[
+ u_N + v_N(A_{N-1}x + B_{N-1}) + w_N[A_{N-1}x^2 + 2A_{N-1}B_{N-1}x + B_{N-1}^2]
\]  

(26)

\[
f_{N-1}(x) = [b_{N-1}(B_{N-1} - d_{N-1})^2 + e_{N-1}B_{N-1}^2 + u_N + v_N B_{N-1} + w_N B_{N-1}^2]
\]

\[
+ x[2b_{N-1}A_{N-1}(B_{N-1} - d_{N-1}) + 2e_{N-1}(A_{N-1} - 1) B_{N-1} + v_N A_{N-1}]
\]

\[
+ 2w_N A_{N-1}B_{N-1}
\]

\[
+ x^2[b_{N-1}A_{N-1}^2 + e_{N-1}(A_{N-1} - 1)^2 + w_N A_{N-1}^2]
\]  

(27)
We may express Eq. (27) as
\[ f_{N-1}(x) = u_{N-1} + v_{N-1}x + w_{N-1}x^2 \] (28)
where
\[ u_{N-1} = b_{N-1}(B_{N-1} - d_{N-1})^2 + e_{N-1}B_{N-1}^2 + u_N + v_NB_{N-1} + w_NB_{N-1}^2 \] (29)
\[ v_{N-1} = 2b_{N-1}A_{N-1}(B_{N-1} - d_{N-1}) + 2e_{N-1}(A_{N-1} - 1)B_{N-1} + v_NA_{N-1} \]
\[ + 2w_NA_{N-1}B_{N-1} \] (30)
\[ w_{N-1} = b_{N-1}A_{N-1}^2 + e_{N-1}(A_{N-1} - 1)^2 + w_NA_{N-1}^2 \] (31)

Equations (29)-(31) relate \( u_{N-1}, v_{N-1}, w_{N-1} \) with \( u_N, v_N, w_N \). Given a starting place \( u_N, v_N, w_N \), it is possible now to generate sequentially the triplet \( u_j, v_j, w_j \). In fact, since \( b_N, d_N, e_N \) are given, \( u_N, v_N, w_N \) can be found via (8) and (9) and (13)-(15).

The sequence of \( x_1, x_2, \ldots, x_N \) may be found starting with
\[ f_1(x) = \operatorname{Min}_{x_1} [b_1(x_1 - d_1)^2 + e_1(x_1 - x)^2 + f_2(x_1)] \] (32)

As shown above in (21)
\[ x_1 = \frac{xe_1 + d_1b_1 - (v_2/2)}{b_1 + e_1 + w_2} \] (33)
where \( v_2 \) and \( w_2 \) are determined from (23), (24), (30), and (31).

The general \( x_k \) is found from (21) since \( b_k, d_k, e_k \), are known for \( k = 1, 2, \ldots, N \):
\[ x_k = \frac{x_1e_k + d_{k-1}b_{k-1} - (v_k/2)}{b_k + e_k + w_{k+1}} \] (34)

It is interesting to compare the discrete quadratic formulation in this section with the continuous case in Chapter 4, Section 23. In both cases functional equations were first developed. These might have been used for computational purposes. Advantage was taken, however, of the quadratic nature of the functional by assuming in both cases a quadratic form for the solution. The equating of coefficients yielded analytical recursion relationships that made it possible to generate numerical results readily.

REFERENCES


**PROBLEMS**

1. Suppose the purchasing cost $c_i$ and the selling price $p_i$ for the warehousing problem described in Sections 2–5 are given for a 10-stage process as

<table>
<thead>
<tr>
<th>$N$:</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_i$:</td>
<td>3</td>
<td>5</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>$c_i$:</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>1</td>
<td>7</td>
<td>6</td>
<td>3</td>
</tr>
</tbody>
</table>
where $B$ is the maximum inventory. Evaluate the solution by the three methods discussed.

2. In the smoothing problem of Section 17 the demand rate is given as

$$
\begin{align*}
  t: & \quad 0 \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7 \quad 8 \quad 9 \quad 10 \\
  r(t): & \quad 0 \quad 10 \quad 5 \quad 3 \quad 3 \quad 5 \quad 12 \quad 2 \quad 5 \quad 7 \quad 10
\end{align*}
$$

Evaluate the minimum cost for $c = 0, 5, 10, 15$. Assume $F(x(t) - r(t)) = x(t) - r(t)$ and $G(dx/dt) = 0.5(dx/dt)^2$.

3. In the smoothing problem of Section 18 let

$$
\begin{align*}
  k: & \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7 \quad 8 \quad 9 \quad 10 \\
  b_k: & \quad 2 \quad 5 \quad 5 \quad 1 \quad 3 \quad 8 \quad 6 \quad 10 \quad 4 \quad 2 \\
  d_k: & \quad 1 \quad 4 \quad 3 \quad 1 \quad 1 \quad 5 \quad 3 \quad 7 \quad 1 \quad 1 \\
  e_k: & \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 4 \quad 4 \quad 3 \quad 5
\end{align*}
$$

Evaluate the minimum cost for $x = 0, 5, 10, 15$.

4. For the in plant inventory problem described in Section 16, let us assume the following

(a) $N$: 1 2 3 4 5 6 7 8 9 10 
$p_i$: 10 5 7 10 10 8 1 3 6 8 
$s_i$: 4 5 6 7 7 3 10 2 5 10

(b) $C_i(h_i) = \bar{h} + 1.5 | (h_i - \bar{h}) |$
$K_i(H_i) = \bar{H} + 2 | (H_i - \bar{H}) |$

(c) $y_i = 1, 2, 3, 4, 5$ with $y_{\text{Max}} = 5$

(d) $m(y_i) = 1 - \frac{y_i}{y_{\text{Max}}}$

(e) $g(y_i) = y_i^2$

(f) $5 \leq h_i \leq 35; \quad 10 \leq H_i \leq 50$

(g) $\bar{h}_i = 20, \quad \bar{H}_i = 30$

(h) The initial levels are

(1) $h = 20, \quad H = 30$ \quad (3) $h = 5, \quad H = 50$
(2) $h = 5, \quad H = 10$ \quad (4) $h = 35, \quad H = 50$

For each of the four initial conditions evaluate $f_N(h, H)$ for $N = 1, 2, ..., 10$ and delineate the infeasible conditions; that is to say, where any choice of the permissible $y_i$ violates condition (f) above.
5. In a work load scheduling of hiring, utilizing, and laying off men, the total cost of the operation is expressed by

\[ C_i = H_i + 120X_i + 200\mu_i(R_i - S_i) + L_i \]

where

\[ C_i = \text{total cost at the } i\text{th week} \]
\[ H_i = \text{hiring cost at the } i\text{th week} \]
\[ S_i = X_i - 0.6(h_i + N_i) \]
\[ N_i = 0.05(R_i - X_i) \]
\[ X_i = \text{total men in plant in the } i\text{th week} \]
\[ R_i = \text{demand for men in the } i\text{th week} \]
\[ h_i = \text{men hired in the } i\text{th week} \]
\[ N_i = \text{transfers in the } i\text{th week} \]
\[ S_i = \text{equivalent number of men to complete work in the } i\text{th week} \]
\[ L_i = \text{cost of layoff in the } i\text{th week} \]

The lay off cost during week \( i \) depends on the number laid off in week \( i \) plus those in week \( i-1 \). If no men are laid off in week \( i \), there is no cost even if men were laid off in week \( i-1 \). See Table 4 for \( L_i \). See Table 5 for demand schedule. Confirm the manpower loading given in Table 6.


**TABLE 4**

**PATTERN FOR LAYOFF COSTS**

<table>
<thead>
<tr>
<th>Number laid off in week ( i, l_i )</th>
<th>Number laid off in week ( i-1, l_{i-1} )</th>
<th>Cost of layoff in week ( i, L_i )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0-700</td>
<td>0</td>
</tr>
<tr>
<td>1-100</td>
<td>0-100</td>
<td>1,000</td>
</tr>
<tr>
<td>0</td>
<td>101-300</td>
<td>6,000</td>
</tr>
<tr>
<td>0</td>
<td>301-500</td>
<td>11,000</td>
</tr>
<tr>
<td>0</td>
<td>501-600</td>
<td>15,000</td>
</tr>
<tr>
<td>101-200</td>
<td>0</td>
<td>1,000</td>
</tr>
<tr>
<td>1-200</td>
<td>1,000</td>
<td></td>
</tr>
<tr>
<td>201-400</td>
<td>11,000</td>
<td></td>
</tr>
<tr>
<td>401-500</td>
<td>15,000</td>
<td></td>
</tr>
<tr>
<td>201-300</td>
<td>15,000</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>101-300</td>
<td></td>
</tr>
<tr>
<td>101-300</td>
<td>25,000</td>
<td></td>
</tr>
<tr>
<td>301-400</td>
<td>35,000</td>
<td></td>
</tr>
<tr>
<td>301-400</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1-200</td>
<td>25,000</td>
<td></td>
</tr>
<tr>
<td>201-300</td>
<td>35,000</td>
<td></td>
</tr>
<tr>
<td>401-500</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>35,000</td>
<td></td>
</tr>
<tr>
<td>101-200</td>
<td>50,000</td>
<td></td>
</tr>
<tr>
<td>501-600</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>35,000</td>
<td></td>
</tr>
<tr>
<td>1-100</td>
<td>50,000</td>
<td></td>
</tr>
<tr>
<td>601-700</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>70,000</td>
<td></td>
</tr>
</tbody>
</table>
6. In a production shop, it is desired to minimize the cost of fabrication of the products by suitable scheduling of the lot sizes to meet the required order. Let

\[ C_i = \text{total expected incremental cost for the } i\text{th lot} \]
\[ S_i = \text{expected initial set up cost for the } i\text{th lot} \]
\[ C_{I,i} = \text{expected inventory cost for the } i\text{th lot} \]
\[ C_{b,i} = \text{expected obsolescence resulting from changes in engineering order for the } i\text{th lot} \]
\[ y_i = \text{size of lot } i \]
\[ Y = \text{total order size} \]
\[ C_i = S_i + C_{I,i} + C_{b,i} = C_i(y_i) \]

The total expected cost for \( N \) lots is

\[ C = \sum_{i=1}^{N} C_i \]
8. INVENTORY PROBLEMS

Since \( Y = \sum_{i=1}^{N} y_i \), show that

\[
    f_N(Y) = \min_{0 \leq y_i \leq Y} \left[ C(y_i) + f_{N-1}(Y - y_i) \right]
\]

where

\[
    f_N(Y) = \text{the minimum cost for } C \text{ over } N \text{ lots to fabricate a total order of size } Y
\]


7. Referring to Sections 13-15, compare the inventory levels for the minimum cost, maximum profit, and maximum rate of return models.

Verify your conclusions with the following data:

\[
    \begin{align*}
    k &= 2 \\
    \xi &= 1 \\
    \delta &= 0.8 \\
    C_0/C_1 &= 0.9 \\
    \mu &= 1 \\
    \eta &= 1.5 \\
    \sigma &= 1 \\
    E &= 0 \\
    \lambda &= 2 \\
    \epsilon &= 1 \\
    p &= 5
    \end{align*}
\]

Repeat the calculations for \( \eta = \epsilon = 1.5 \).

8. Referring to Sections 13-15, compare the inventory levels for the minimum cost, maximum profit, and maximum rate of return models for the density function \( \varphi(s) = e^{-s} \).

Verify your conclusion with the following data:

\[
    \begin{align*}
    k &= 2 \\
    \xi &= 1 \\
    \delta &= 0.8 \\
    C_0/C_1 &= 0.9 \\
    \mu &= 1 \\
    \eta &= 1.5 \\
    \sigma &= 1 \\
    E &= 0 \\
    \lambda &= 2 \\
    \epsilon &= 1 \\
    p &= 5
    \end{align*}
\]

How does one test the data for consistency in these models?

Given the following data:

\[
    \begin{align*}
    k &= 2 \\
    \xi &= 1 \\
    \delta &= 0.8 \\
    \sigma &= 1 \\
    \mu &= 1 \\
    \eta &= 1 \\
    p &= 5 \\
    E &= 0 \\
    \lambda &= 1 \\
    \epsilon &= 3 \\
    \frac{C_0}{C_1} &= 0.9
    \end{align*}
\]

do solutions exist for these models for \( \varphi(s) = e^{-s} \) and for

\[
    \varphi(s) = \frac{1}{\sigma\sqrt{2\pi}} \exp \left[ -\frac{1}{2} \left( \frac{s-E}{\sigma} \right)^2 \right]?
\]

9. On the assumption that a unique solution exists for the inventory problem in Section 9, where the penalty cost is \( py + q \), the minimum inventory cost is given by

\[
    f(x) = k(\bar{x} - x) + a \left[ p \int_{x}^{\infty} (s - \bar{x}) \varphi(s) \, ds + f(0)\int_{x}^{\infty} \varphi(s) \, ds \right] \\
    + \int_{0}^{x} f(\bar{x} - s) \varphi(s) \, ds + q \int_{x}^{\infty} \varphi(s) \, ds
\]

(1)
where \( \bar{x} \) is the only positive root of

\[
G(y) = k - a \left[ p \int_{y}^{\infty} \varphi(s) \, ds + q \varphi(y) + k \int_{0}^{y} \varphi(s) \, ds \right] = 0
\] (2)

If the penalty cost is assumed incorrectly to be \( p' y + q \) rather than \( py + q \), the inventory cost is

\[
f_{p'}(x) = k(\bar{x} - x) + a \left[ p' \int_{\bar{x}}^{\infty} (s - \bar{x}) \varphi(s) \, ds + f_{p'}(0) \int_{\bar{x}}^{\infty} \varphi(s) \, ds \\
+ \int_{0}^{\bar{x}'} f_{p'}(\bar{x}' - s) \varphi(s) \, ds + q \int_{\bar{x}'}^{\infty} \varphi(s) \, ds \right]
\] (3)

where \( \bar{x}' \) is the root of

\[
k - a \left[ p' \int_{\bar{x}'}^{\infty} \varphi(s) \, ds + q \varphi(\bar{x}') + k \int_{0}^{\bar{x}'} \varphi(s) \, ds \right] = 0
\] (4)

Note that \( f(x) \) is the inventory cost associated with \( py + q \) and \( f_{p'}(x) \) with \( p'y + q \). Using the following

\[
f_{p'}(x) = -k, \quad 0 \leq x \leq \bar{x}'
\] (5)
\[
f'(x) = -k, \quad 0 \leq x \leq \bar{x}
\] (6)

and

\[
f_{p'}(x) = f_{p'}(0) - kx
\] (7)
\[
f(x) = f(0) - kx
\] (8)

show that

\[
f(0) = \frac{1}{1 - a} \left[ k\bar{x} + a \left\{ p' \int_{\bar{x}}^{\infty} (s - \bar{x}) \varphi(s) \, ds - k \int_{0}^{\bar{x}} (\bar{x} - s) \varphi(s) \, ds \\
+ q \int_{\bar{x}}^{\infty} \varphi(s) \, ds \right\} \right]
\] (9)
\[
f_{p'}(0) = \frac{1}{1 - a} \left[ k\bar{x}' + a \left\{ p' \int_{\bar{x}'}^{\infty} (s - \bar{x}') \varphi(s) \, ds \\
- k \int_{0}^{\bar{x}'} (\bar{x}' - s) \varphi(s) \, ds + q \int_{\bar{x}'}^{\infty} \varphi(s) \, ds \right\} \right]
\] (10)

A comparison of (9) and (10) yield the relative cost of an incorrect \( p \).

(Gluss, B., Costs of incorrect data in optimal inventory computation. Management Sci. 6, No. 4, 491–497 (1960).)
10. Show that Eqs. (2) and (4) in the previous problem may be written as

\[ k - ap + a(p - k) \int_0^{\tilde{x}} \varphi(s) \, ds - aq \varphi(\tilde{x}) = 0 \]

\[ k - ap' + a(p' - k) \int_0^{\tilde{x}'} \varphi(s) \, ds - aq \varphi(\tilde{x}') = 0 \]

Using

\[ u = \frac{\int_{\tilde{x}}^{\tilde{x}'} s \varphi(s) \, ds}{\int_{\tilde{x}}^{\tilde{x}'} \varphi(s) \, ds} \quad \text{where} \quad \begin{cases} \tilde{x} < u < \tilde{x}' \text{, or} \\ \tilde{x}' < u < \tilde{x} \end{cases} \]

show that

\[ f_p(x) - f(x) = \frac{k(p' - p) (\tilde{x}' - u)}{p' - k} - \frac{aq \varphi(\tilde{x}')}{(1 - a)(p' - k)} \]

\[ \times [q - (\tilde{x}' - u)(p - k)] + \frac{aq \varphi(\tilde{x})}{(1 - a)(p - k)} \]

\[ \times [q + (u - \tilde{x})(p - k)] - \frac{kq(p' - p)}{(p - k)(p' - k)} \]

11. If the penalty cost is assumed incorrectly to be \( py + q' \) instead of \( py + q \), show that

\[ f_q(x) - f(x) = \frac{a}{1 - a} \left[ q'(\tilde{x}' - u) \varphi(\tilde{x}') + q(u - \tilde{x}) \varphi(\tilde{x}) \right. \]

\[ - \frac{aq}{p - k} (q' \varphi(\tilde{x}') - q \varphi(\tilde{x})) \]

where \( \tilde{x}' \) satisfies

\[ k - ap + a(p - k) \int_0^{\tilde{x}'} \varphi(s) \, ds - aq \varphi(\tilde{x}') = 0 \]

Here \( f_q(x) \) is the minimum inventory cost associated with the incorrect penalty cost \( py + q' \).

If the incorrect discount ratio \( a' \) is used instead of \( a \), show that the relative cost of the incorrect discount ratio is

\[ f_{a'}(x) - f(x) = \frac{1}{1 - a} \left[ k(\tilde{x}' - u) \left( 1 - \frac{a}{a'} \right) - kq \left( 1 - \frac{a}{a'} \right) \right. \]

\[ + \frac{aq}{p - k} \left( \frac{q}{p - k} \varphi(\tilde{x}') + \left( u - \tilde{x} + \frac{q}{p - k} \right) \varphi(\tilde{x}) \right] \]
where $x'$ satisfies
\[ k - a'p + a'(p - k) \int_0^{x'} q(s) \, ds - a'q'q(x') = 0 \]

The $f_a(x)$ is the minimum inventory cost associated with the incorrect discount ratio $a'$.

12. If for Problems 9-11 we have the following data:

\[ \varphi(s) = \frac{1}{4} se^{-s/2}, \quad 0 \leq y < \infty \]
\[ p = 2k, \quad p' = 2.5k \]
\[ q = 10k, \quad q' = 11k \]
\[ a = 0.97, \quad a' = 0.96 \]

show that
\[ \tilde{x} = 14.5499, \quad f(x) = -kx + 146.1519 \]
for $p'$, \[ \tilde{x}' = 14.7542, \quad \frac{f_a(x) - f(x)}{f(x)} = \frac{0.0091}{146.1519 - x} \]
for $q'$, \[ \tilde{x}' = 14.7315, \quad \frac{f_a(x) - f(x)}{f(x)} = \frac{0.1451}{146.1519 - x} \]
for $a'$, \[ \tilde{x}' = 13.8587, \quad \frac{f_a(x) - f(x)}{f(x)} = \frac{0.1144}{146.1519 - x} \]

Discuss the significance of these results

13. Set up the functional equations of dynamic programming to maximize over $y(t)$ a plant expansion program given by
\[ \int_0^T \left[ \min \{ \epsilon(t), y(t) \} - c \frac{dy}{dt} \right] e^{-rt} \, dt \]
where
\[ y(t) = \text{capacity at time } t \]
\[ dy/dt = \text{rate of expansion} \]
\[ \epsilon(t) = \text{demand at time } t \]
\[ c = \text{unit cost of new capacity} \]

Assume that $y(t)$ is monotone increasing and $y(0)$ is given.

(Arrow, K. J., Karlin, S., and Scarf, H. [2].)

14. Given the inventory equation from Section 9

\[ F(y) = k y + a \left[ p \int_y^\infty (s - y) \varphi(s) \, ds + (f(0) + q) \int_y^\infty \varphi(s) \, ds \right. \]
\[ \left. + \int_0^y f(y - s) \varphi(s) \, ds \right] \]
It has been shown by Bellman that $\varphi'(s) \leq 0$ for all $s$ is a sufficient condition to ensure the simple ordering policy.

(a) increase stock to $x$, $0 \leq x \leq \bar{x}$
(b) order nothing, $x \geq \bar{x}$

We will be asked to show that less stringent requirements on $\varphi(x)$, namely that $\varphi(x)$ be a Pearson distribution, also yield the simple ordering policy.

Given the following that

(a) all relative minima of $F(y)$ satisfies

$$G(y) = (1 - a)k - a(p - k) \int_{y}^{\infty} \varphi(s) ds - aq\varphi(y) = 0$$

(b) the roots of (2) are separated by those of

$$\frac{dG}{dy} = a(p - k) \varphi(y) - aq\varphi'(y) = 0$$

or rearranging

$$\frac{\varphi'(y)}{\varphi(y)} = \frac{p - k}{q} = A$$

(c) assuming that $ap > k - aq\varphi(0)$ or equivalently $ap > k$, then $G(0) < 0$

Let

$$\frac{\varphi'(y)}{\varphi(y)} = \frac{L(y)}{M(y)}, \text{ where } L(y) \text{ and } M(y) \text{ are quadratic functions of } y$$

$$L(y) = AM(y)$$

When $L(y)$ is linear, $\varphi(y)$ is Pearsonian by definition. By examining the roots of Eq. (6) determine the conditions under which (2) has only one root. In particular, show that (2) has only one root, namely $\bar{x}$, when

(a) there is only one positive root to Eq. (6); $y_1$
(b) there are two positive roots of (6) and $G''(0) < 0$;
(c) there are two positive roots of (6), $G'(0) > 0$ and $G(y_2) > 0$ where $y_1$ and $y_2$ are roots of (6).

Sketch $G(y)$ vs. $y$. Use the fact that

$$G(0) = -(ap - k) - aq\varphi(0) < 0; \quad G(\infty) = (1 - a)k > 0$$

(Gluss, B., An optimal inventory solution for some specific demand distributions. *Naval Research Logistics Quart.* 7, No. 1 (1960).)
15. Using chi-square distribution, determine the optimal level in Problem 14:

\[ \varphi(y) = a \exp \left( -\frac{y}{2} \right) y^{(n/2) - 1}, \quad 0 \leq y < \infty \]

\[ \frac{\varphi'(y)}{\varphi(y)} = \frac{n - 2 - y}{2y} \]

Show \( y_1 = \frac{n - 2}{2A + 1} \). Show \( x = 7.848 \) using \( n = 12, p = 2k, q = 10k, a = 0.97 \).

16. Let us consider a sales-warehousing problem in which we desire to minimize the expected penalty costs associated with preventing stock-outs. Let

\[ S_i = \text{sales demand at } i^{th} \text{ warehouse} \]
\[ Q_i = \text{quantity to be allocated to } i^{th} \text{ warehouse} \]
\[ a_i = \text{minimum inventory at } i^{th} \text{ warehouse} \]
\[ P = \text{probability that sales at the } i^{th} \text{ warehouse are greater than or equal to } (Q_i - a_i); \quad P(S_i > Q_i - a_i) \]
\[ p_i = \text{penalty at the } i^{th} \text{ warehouse if } S_i > Q_i - a_i \]

We assume that there is normal replenishment ordering cycle. We desire to minimize the running out of stocks between order points. The model has two possibilities:

(a) Sales at the \( i^{th} \) warehouse until the next replenishment equal or exceed the quantity \( (Q_i - a_i) \).

(b) Sales until the next replenishment are less than \( (Q_i - a_i) \).

If the level in the warehouse gets down to the emergency level \( a_i \), we must make an emergency order. Show that the total expected cost is

\[ \sum_{i=1}^{i=N} p_i p(S_i \geq Q_i - a_i) \]

where

\[ Q_i \geq 0 \quad \text{and} \quad \sum_{i=1}^{N} Q_i = Q \]

Prove the theorem: A necessary condition that an allocation has a minimum total expected cost is that the weighted probability \( p_i P(S_i = Q_i - a_i) \) be equal for all warehouses. Here \( p_i \) is the penalty associated with the emergency replenishment action, and \( P(S_i = Q_i - a_i) \) is the probability that sales will be exactly equal to the quantity allocated minus the emergency trigger level.

(Simpson, K. F., Jr., Theory of allocation of stocks to warehouses. Operations Research 7, No. 6, 797–806 (1959).)
17. Referring to Problem 16 suppose now no emergency replacements are permitted; that is to say, if the warehouse runs out, it is not replenished until the normal replenishment period. We define

\[ w_i = \text{penalty of lost sales at the } i\text{th warehouse} \]
\[ t = \text{sales demand} \]
\[ d_i(t) = \text{density of sales function} \]

The penalty at the \( i\)th warehouse for sales demand \( t \) is

\[ p_i(t) = \text{Max} \{0, w_i(t - Q_i)\} \]

Probability that sales are within the range \( t \) to \( t + dt \) is

\[ P(t < S_i < t + dt) = d_i(t)dt \]

Show that the expected penalty at \( i\)th depot is

\[ \int_0^\infty p_i(t) d_i(t) dt = w_i \int_{Q_i}^\infty (t - Q_i) d_i(t) dt \]

and over all depots the total expected penalty is

\[ \sum_{i=1}^N \int_{Q_i}^\infty (t - Q_i) d_i(t) dt \]

subject to

\[ \sum_{i=1}^N Q_i = Q \]

Set up the functional equations by dynamic programming where optimization is over the set of \( Q_i \).

Prove the following theorem: A necessary condition that an allocation minimizes the weighted number of lost sales is that the weighted probabilities \( w_i P(S_i \geq Q_i) \) be equal for all warehouses. Here \( w_i \) is the weight given to a lost customer at a particular warehouse and \( P(S_i \geq Q_i) \) is the probability that demand is equal to or greater than the quantity allocated that warehouse.

18. In a planning model it is desired to provide capital equipment installations to meet a service demand for a time span of \( N \) years. It is assumed that the growth of demand curve is exponential. The cumulative demand up to time \( t \) is \( \exp(kt) \). The cost of installing the equipment is \( b \) and the cost of the equipment itself is \( a \). The interest cost of money tied up in equipment is \( c \) per cent per year. The term \( a \exp kt \) is the cost of the equipment to meet the growing demand. Note the cost of interest \( (1 + c)^{-r-s} \approx \exp(-c(r-s)) \).

If we define

\[ f_i(s) = \text{the minimal cost of planning for the demand period from } s \text{ to } N \]

in at most \( t \) installments, beginning with an installment at time \( s \).
Show that
\[ f_t(s) = \min_{s \leq r \leq N} \left[ b + a(\exp(kr) - \exp(ks)) + \exp\{-(r - s)\} f_{t-1}(r) \right] \]


19. Let us consider an inventory problem with the following costs: an inventory acquisition, inventory holding, and backlog:

- \( i_0 = \) initial inventory
- \( i_1 = \) inventory one stage later
- \( \alpha = \) present worth factor
- \( \beta = 1 - \alpha \)
- \( x = \) random order variable
- \( \mu = \) average \( x \)
- \( c_2 = \) cost of acquisition per unit
- \( q_2 = \) quantity ordered
- \( h = \) holding cost per unit
- \( k = \) backlog cost per unit

The expected value of the inventory acquisition is approximated by

\[ V_a = c_2(i_0 + i_1 + q_2) \left( 1 - \frac{\beta(i_0 + i_1 + q_2)}{\mu} \right) \]

The expected inventory holding cost is

\[ V_h = -h \left[ \frac{(i_0 + i_1 + q_2)^2}{2\mu} - i_1 - 2q_2 \right] \]

The expected number of backlogs for the current, next, and second periods are given by

\[ \int_0^\infty \max(0, x - i_0) f(x) \, dx, \quad \int_0^\infty \max(0, y - (i_0 + i_1 + q_2) g(y) \, dy, \]

\[ \int_0^\infty \max(0, z - i_0 - i_1 - q_2 h(z) \, dz \]

where the frequency demands for \( x, y, z \) for the next, next two, and next three periods are \( f(x), g(y), h(z) \), respectively. Discounting linearly, the expected backlog cost is

\[ V_b = -k \int_0^\infty (x - i_0) f(x) \, dx - k(1 - \beta) \int_{t_0 + t_1}^\infty (y - i_0 - i_1) g(y) \, dy \]

\[ - k(1 - 2\beta) \int_{t_0 + t_1 + q_2}^\infty (z - i_0 - i_1 - q_2) h(z) \, dz \]
Totaling we have

\[ V(i_0, i_1, q_2) = c_2 i_0 + (c_2 + h) i_1 + (c_2 + 2h) q_2 \]

\[ - \frac{h + 2\beta c_2}{2\mu} (i_0 + i_1 + q_2)^2 - k \int_{i_0}^{\infty} (x - i_0) f(x) \, dx \]

\[ - k(1 - \beta) \int_{i_1}^{\infty} (y - i_0 - i_1) g(y) \, dy \]

\[ - k(1 - 2\beta) \int_{i_0 + i_1 + q_2}^{\infty} (x - i_0 - i_1 - q_2) h(x) \, dx \]

Our objective is to determine the maximum inventory valuation \( F(s) \), where \( s = (i_0, i_1) \):

\[ F(s) = \max_{q_2 \geq 0} [V(i_0, i_1, q_2) - c_2 q_2] \]

Show that when \( q_2 > 0 \)

\[ \int_{i_0 + i_1 + q_2}^{\infty} h(x) \, dx = \frac{h}{\mu k(1 - 2\beta)} \left[ i_0 + i_1 + q_2 - 2\mu + \frac{2\beta c_2}{h} (i_0 + i_1 + q_2) \right] \]

Discuss the solution of this equation.

(Mills, H. D., Inventory valuations—An analytic technique. Management Sci. 8, No. 1, 58–63 (1961).)

20. For a two-commodity inventory problem corresponding to that in Sections 6 and 7, justify

\[ f(x_1, x_2) = \min_{y_1 \geq x_1, y_2 \geq x_2} \left[ k_1(y_1 - x_1) + k_2(y_2 - x_2) \right] \]

\[ + a \left\{ \int_{y_1}^{\infty} \int_{y_2}^{\infty} \left[ p_1(s_1 - y_1) + p_2(s_2 - y_2) + f(0, 0) \right] dG(s_1, s_2) \right\} \]

\[ + \int_{y_1}^{\infty} \int_{y_2}^{\infty} \left[ p_1(s_1 - y_1) + f(0, y_2 - s_2) \right] dG(s_1, s_2) \]

\[ + \int_{y_1}^{\infty} \int_{y_2}^{\infty} \left[ f(y_1 - s_1, 0) + p_2(s_2 - y_2) \right] dG(s_1, s_2) \]

\[ + \int_{y_1}^{\infty} \int_{y_2}^{\infty} f(y_1 - s_1, y_2 - s_2) dG(s_1, s_2) \]
Indicate by block diagram the inventory situation corresponding to each term.

Assuming that

(a) \( k_i(y) = k_i y, \quad p_i(y) = p_i y; \quad k_i, p_i > 0 \)

(b) \( dG(s_1, s_2) = \varphi(s_1, s_2) \, ds_1 \, ds_2; \quad \varphi > 0 \)

(c) \( \int_0^\infty \int_0^\infty s_i \varphi(s_1, s_2) \, ds_1 \, ds_2 < \infty; \quad i = 1, 2 \)

(d) \( 0 < a < 1; \)

(e) \( \alpha p_i > k_i; \quad i = 1, 2 \)

Show that

\[ y_i = \bar{x}_i, \quad 0 \leq x_i \leq \bar{x}_i; \quad i = 1, 2 \]

\[ y_i = x_i, \quad x_i > \bar{x}_i; \quad i = 1, 2 \]

where \( x_1 \) and \( x_2 \) are the roots, respectively, of

\[ k_1 - p_1 a \int_0^\infty \left( \int_0^\infty \varphi(s_1, s_2) \, ds_2 \right) \, ds_1 = 0 \]

\[ k_2 - p_2 a \int_0^\infty \left( \int_0^\infty \varphi(s_1, s_2) \, ds_2 \right) \, ds_1 = 0 \]

Observe that the policy is independent of the dimensionality of the problem.


21. Given a one-stage inventory model and the cost expression

\[ L(y) = (y - x) \cdot c + K(y - x) + h(y - \varepsilon) + p(\varepsilon - y) - v(y - \varepsilon) - r \cdot \text{Min} (\varepsilon, y) \]

where

\( \varepsilon = \text{the demand} \)

\( \varphi(\varepsilon) = \text{demand density function} \)

\( h(x) = \text{holding cost} \)

\( p(x) = \text{penalty cost} \)

\( v(x) = \text{salvage value} \)

\( E = \text{expectation} \)

\( r = \text{unit sales price} \)

\( c, K = \text{positive constants in ordering cost} = cz + K_2 \)

\( x = \text{current inventory level} \)

\( y = \text{new inventory level} \)

\( z = y - x = \text{order quantity}; z \geq 0 \)

The expected cost is

\[ EL(y) = -cx + K(y - x) + G(y) \]
where
\[ G(y) = c \cdot y + \int_0^y [h(y - \epsilon) - v(y - \epsilon) - r \cdot \epsilon] \varphi(\epsilon) \, d\epsilon \]
\[ + \int_y^\infty [p(\epsilon - y) - r \cdot y] \varphi(\epsilon) \, d\epsilon \]

Show that the first and second derivatives of \( G(y) \) are
\[ G'(y) = c + \int_0^y [h'(y - \epsilon) - v'(y - \epsilon)] \varphi(\epsilon) \, d\epsilon \]
\[ - \int_y^\infty [p'(\epsilon - y) + r] \varphi(\epsilon) \, d\epsilon \]
\[ G''(y) = \int_0^y [h''(y - \epsilon) - v''(y - \epsilon)] \varphi(\epsilon) \, d\epsilon \]
\[ + \int_y^\infty p''(\epsilon - y) \varphi(\epsilon) \, d\epsilon + [p'(0) + h'(0) - v'(0) + r] \varphi(0) \]

Show that for \( K = 0 \) that the simple ordering rule: if \( x < y_0 \), order up to \( y_0 \), if \( x > y_0 \), do not order; applies if \( G(y) \) has a unique minimum at \( y_0 \). This occurs if

(a) \( E \varphi' > c - r \) \([G(0) \text{ is not a relative minimum}]\)
(b) \( G(y) \to \infty \) as \( y \to \infty \)
(c) \( G'(y) = 0 \) has a unique solution

(Arrow, K. J., Karlin, S., and Scarf, H. [2].)

22. For the inventory model in Problem 21 show the simple ordering policy holds if \([h(y - \epsilon) - v(y - \epsilon)]\) and \(p(\epsilon - y)\) are increasing convex functions. Use the facts that \( G(y) \) must be convex and hence \( G''(y) > 0 \) for all \( y \geq 0 \) and \( G'(y) \) has one root, and use the conditions (a) and (b) of the previous problem.

(Arrow, K. J., Karlin, S., and Scarf, H. [2].)

23. Given
\[ f(x) = \min_{y \geq x} [-cx + G(y)] \]

where
\[ G(y) = cy + \int_0^y [h(y - \epsilon) - r \cdot \epsilon] \varphi(\epsilon) \, d\epsilon + \int_y^\infty [p(\epsilon - y) - r \cdot y] \varphi(\epsilon) \, d\epsilon \]
\[ + \alpha \left[ f(0) \int_0^y \varphi(\epsilon) \, d\epsilon + \int_y^\infty f(y - \epsilon) \varphi(\epsilon) \, d\epsilon \right] \]
All the variables are defined in Problem 21, except $\alpha$ which is the discount factor.

Let

$$M(x) = c + \int_0^x [h'(x - \epsilon) - \alpha c] \phi(\epsilon) d\epsilon - \int_x^\infty [r + p'(\epsilon - x)] \phi(\epsilon) d\epsilon$$

and

$$\int_0^\infty p'(\epsilon) \phi(\epsilon) d\epsilon = E(p') > c - r$$

where $h'$ and $p'$ are derivatives.

Given the following statements that

$$G'(x) \geq M(x) \quad \text{for all } x; \quad G'(x) = M(x) \quad \text{for } x \leq x^*$$

where $x^*$ is the optimal stock level.

Prove that if $M(x)$ has a unique zero at $x = x^*$, then the optimal ordering policy has the simple form:

$$x < x^*, \quad \text{order up to } x^*; \quad x > x^*, \quad \text{do not order}$$

(Arrow, K. J., Karlin, S., and Scarf, H. [2].)

24. In Problem 23 what is the significance of the assumption

$$\int_0^\infty p'(\epsilon) \phi(\epsilon) d\epsilon = E(p') > c - r$$

Under what circumstances is this assumption true?

(Arrow, K. J., Karlin, S., and Scarf, H. [2].)

25. In Problem 23 show that $M(x)$ is strictly increasing and therefore has a unique zero, if $h(t)$ and $p(t)$ are convex, and $h'(0) + p'(0) + r > \alpha c$

(Arrow, K. J., Karlin, S., and Scarf, H. [2].)

26. For Problem 23, given

$$h(y - \epsilon) = (y - \epsilon)^2, \quad p(y - \epsilon) = e^{(y - \epsilon)};$$

$$r = 2, \quad c = 1, \quad \alpha = 0$$

(a) $\phi(\epsilon) = \frac{1}{10} \exp - \left(\frac{\epsilon}{10}\right)$,

(b) $\phi(\epsilon) = \frac{1}{10 \sqrt{2\pi}} \exp \left[- \left(\frac{\epsilon}{10}\right)^2 / 2\right]$,

find $x^*$. 


27. For a single commodity problem corresponding to Section 7, if the distribution demand \( g(s) \) is not the same in successive periods, show that the cost is given by

\[
f_i(x) = \min_{y \geq x} \left[ k(y - x) + a \int_y^\infty p(s - y) g_i(s) ds + f_{i+1}(0) \int_y^\infty g_i(s) ds \right. \\
\left. + \int_0^y f_{i+1}(y - s) g_i(s) ds \right]
\]  

We define the expression "demand is increasing" by the following statement regarding the distribution function, namely,

\[
\int_0^x g_{i+1}(s) ds \leq \int_0^x g_i(s) ds
\]

With demand increasing, this implies that the optimizing value of \( y \) cannot be smaller than its value if \( g_i(s) = g_{i+1}(s) \). Now, if the stocks left at the end of one period are not greater than optimal \( y \) required for the succeeding period, then we can evaluate the left over stocks at their purchase price. Under this condition, justify the following equation for time period \( i \):

\[
c_i(x) = \min_{y \geq x} \left[ k(y - x) + a \int_y^\infty p(s - y) g_i(s) ds + \int_0^y k(y - s) g_i(s) ds \right]
\]

Show that

\[
\frac{dM(y)}{dy} = k - ap \int_y^\infty g_i(s) ds - ak \int_0^y g_i(s) ds
\]

Noting that \( M(y) \) is convex and \( dM(y)/dy = 0 \), show that

\[
\int_0^y g_i(s) ds = \frac{(ap - k)}{a(p - k)}
\]

which determines the optimal \( y = \bar{x} \). Show that the optimal policy is

- order \( \bar{x} - x \) if \( x < \bar{x} \)
- do not order if \( x \geq \bar{x} \)

(Levy, J. Optimal inventory policy when demand is increasing. Operations Research 8, No. 6, 861–863 (1960).)
28. Suppose we have a time lag inventory problem where \( \lambda \) is the total lag time. We let \( y_1, y_2, \ldots, y_{\lambda-1} \) represent the outstanding orders where \( y_1 \) is due in one time period, hence \( y_2 \) is due in two time periods hence, etc. Define

- \( x = \) current stock size
- \( \epsilon = \) demand
- \( \varphi(\epsilon) = \) demand distribution
- \( h(x) = \) holding cost
- \( p(x) = \) penalty cost
- \( c(\epsilon) = \) cost of ordering
- \( a = \) discount factor

If demand exceeds supply, and if we consider this as lost sales, show that the functional equation is

\[
f(x, y_1, y_2, \ldots, y_{\lambda-1}) = \min_{x \geq 0} \left[ c(\epsilon) + \int_0^x h(x - \epsilon) \varphi(\epsilon) \, d\epsilon \right.
\]

\[
+ \int_x^\infty p(\epsilon - x) \varphi(\epsilon) \, d\epsilon
\]

\[
+ \alpha \int_0^x f(x - \epsilon + y_1, y_2, \ldots, y_{\lambda-1}, \epsilon) \varphi(\epsilon) \, d\epsilon
\]

(Arrow, K. J., Karlin, S., and Scarf, H. [2].)

29. Referring to Problem 28 if we permit excess demand to be fulfilled at a later time, thus permitting the stock level \( x \) to assume both positive and negative values (where negative \( x \) means, of course, material that is owed), show that the functional equations are, for \( x > 0 \),

\[
f(x, y_1, y_2, \ldots, y_{\lambda-1}) = \min_{x \geq 0} \left[ c(\epsilon) + \int_0^x h(x - \epsilon) \varphi(\epsilon) \, d\epsilon \right.
\]

\[
+ \int_x^\infty p(\epsilon - x) \varphi(\epsilon) \, d\epsilon
\]

\[
+ \alpha \int_0^x f(x - \epsilon + y_1, y_2, \ldots, y_{\lambda-1}, \epsilon) \varphi(\epsilon) \, d\epsilon
\]

and for \( x < 0 \)

\[
f(x, y_1, y_2, \ldots, y_{\lambda-1}) = \min_{x \geq 0} \left[ c(\epsilon) + \int_0^\infty p(\epsilon - x) \varphi(\epsilon) \, d\epsilon \right.
\]

\[
+ \alpha \int_x^\infty f(x - \epsilon + y_1, y_2, \ldots, y_{\lambda-1}, \epsilon) \varphi(\epsilon) \, d\epsilon
\]

30. Prove that the solution for the deferred payment of demand model in Problem 29 has the form

\[
z^* = z^*(x + y_1 + y_2 + \ldots + y_{\lambda-1})
\]
while the solution for the lost sales model (Problem 28) has for \( \lambda = 1 \) and for all linear costs the form

\[
\begin{align*}
  z^*(x) &> 0, & x < \bar{x} \\
  z^*(x) &= 0, & x \geq \bar{x}
\end{align*}
\]

(Arrow, K. J., Karlin, S., and Scarf, H. [2].)

31. For the deferred payment model in Problem 29 for a single stage lag, \( \lambda = 1 \), show that the functional equation is for an \( n \)-stage process

\[
f_n(x) = \min_{z \geq 0} \left[ cz + L(x) + \alpha \int_0^\infty f_{n-1}(x + z - \varepsilon) \varphi(\varepsilon) \, d\varepsilon \right]
\]

where

\[
L(x) = \begin{cases} 
  h(x - \varepsilon) \varphi(\varepsilon) \, d\varepsilon + \int_x^\infty p(\varepsilon - x) \varphi(\varepsilon) \, d\varepsilon, & x > 0 \\
  p(x) \varphi(\varepsilon) \, d\varepsilon, & x < 0
\end{cases}
\]

(Arrow, K. J., Karlin, S., and Scarf, H. [2].)

32. For the previous problem we take the following:

(a) \( \lim_{z \to \infty} p'(x) > \frac{1 - \alpha}{\alpha} c \);
(b) \( \bar{x}_n \geq \bar{x}_{n-1} \)
(c) \( f_n'(x) = \begin{cases} 
  -c + L'(x), & x < \bar{x}_n \\
  L'(x) + \int_0^\infty f_{n-1}'(x - \varepsilon) \varphi(\varepsilon) \, d\varepsilon, & x > \bar{x}_n
\end{cases} \)
(d) \( f_n(x) \) is convex and \( f_n''(x) \) exists everywhere
(e) \( f_n'(x) \geq f_n'(x) \) for all \( x \)
(f) \( z^*(x) = \begin{cases} 
  \bar{x}_n - x, & x < \bar{x}_n \\
  0, & x \geq \bar{x}_n
\end{cases} \)

where \( \bar{x}_n \) is the unique root of

\[
c + \alpha \int_0^\infty f_{n-1}'(\bar{x}_n - \varepsilon) \varphi(\varepsilon) \, d\varepsilon = 0
\]

Show the properties (a) to (f) hold for \( z^*_{n+1}(x) \).

(Arrow, K. J., Karlin, S., and Scarf, H. [2].)
33. Extending the results of the previous example for $n \to \infty$, show for the continuous case that

$$z^*(x) = \begin{cases} \hat{x} - x, & x < \hat{x} \\ 0, & x \geq \hat{x} \end{cases}$$

where $\hat{x}$ is the root of

$$c + \alpha \int_0^\infty f'(x - \epsilon) \varphi(\epsilon) \, d\epsilon = 0$$

and $f$ is convex. Using the value of $\hat{x}$ in the functional equation we have

$$f(x) = c(\hat{x} - x) + L(x) + \alpha \int_0^\infty f(\hat{x} - \epsilon) \varphi(\epsilon) \, d\epsilon$$

Since $L(x)$ is known, show that an alternate equation for finding $\hat{x}$ is

$$c(1 - \alpha) + \alpha \int_0^\infty L'(x - \epsilon) \varphi(\epsilon) \, d\epsilon = 0$$

Here $\hat{x}$ is the unique root.

34. For a two-stage lag deferred payment of demand model, we have

$$f_n(x, y_1) = \min_{\epsilon \geq 0} \left[ c \cdot x + L(x) + \alpha \int_0^\infty f_{n-1}(x + y_1 - \epsilon, z) \varphi(\epsilon) \, d\epsilon \right]$$

Let us assume the following:

(a) $f_{n-1}(x, y_1) = L(x) + b_{n-1}(x + y_1)$

(b) $z^*_{n-1}(x, y_1) = \bar{x}_{n-1} - x - y_1, \quad x + y_1 < \bar{x}_{n-1}$

(c) $\bar{x}_{n-1}$ is unique root of

$$c + \alpha \int_0^\infty b'_{n-2}(w - \epsilon) \varphi(\epsilon) \, d\epsilon = 0,$$

where $w = x + y_1 + \pi$

(d) $\bar{x}_{n-1} \geq \bar{x}_{n-2}$

(e) $b'_{n-1}(x + y_1) = \begin{cases} -c + \alpha \int_0^\infty L'(x + y_1 - \epsilon) \varphi(\epsilon) \, d\epsilon, & y + x < \bar{x}_{n-1} \\ \alpha \int_0^\infty L'(x + y_1 - \epsilon) + \alpha \int_0^\infty b'_{n-2}(x + y_1 - \epsilon) \varphi(\epsilon) \, d\epsilon, & y_1 + x \geq \bar{x}_{n-1} \end{cases}$

(f) $b_{n-1}(u)$ is convex in $u$; $b''_{n-1}(u) > 0$ for all $u$ except $u = \bar{x}_{n-1}$

where right-and left-hand derivatives exist.

(g) $b'_{n-2}(u) \geq b'_{n-1}(u)$ for all $u$. 
Show that the optimal policy is
\[ z^*(x, y_1) = \begin{cases} \bar{x} - x - y_1, & x + y_1 < \bar{x} \\ 0, & x + y_1 \geq \bar{x} \end{cases} \]

Show that \( \bar{x} \) is the unique root of
\[ c(1 - \alpha) + \alpha^2 \int_0^\infty \int_0^\infty L'(\bar{x} - \epsilon - \eta) \varphi(\epsilon) \varphi(\eta) \, d\epsilon \, d\eta = 0 \]

(Arrow, K. J., Karlin, S., and Scarf, H. [2].)

35. For the general \( \lambda \) lag deferred payment of demand model, deduce
\[
f(x, y_1, y_2, ..., y_{\lambda-1}) = L(x) + L_1(x + y_1) + L_2(x + y_1 + y_2) + ... \\
+ L_{\lambda-2}(x + y_1 + y_2 + ... + y_{\lambda-2}) \\
+ b(x + y_1 + y_2 + ... + y_\lambda)
\]
and
\[ z^*(x, y_1, y_2, ..., y_{\lambda-1}) = \begin{cases} \bar{x} - (x + y_1 + y_2 + ... + y_{\lambda-1}), & x + y_1 + y_2 + ... + y_{\lambda-1} < \bar{x} \\ 0, & \text{otherwise} \end{cases} \]

where
\[
b'(w) = \begin{cases} -\epsilon + \alpha \int_0^\infty L'_{\lambda-2}(w - \epsilon) \varphi(\epsilon) \, d\epsilon, & w < \bar{x} \\ L'_{\lambda-2}(w - \epsilon) + \alpha \int_0^\infty b'(w - \epsilon) \varphi(\epsilon) \, d\epsilon, & w > \bar{x} \end{cases}
\]
and
\[ L_r(u) = \int_0^\infty L_{r-1}(u - \epsilon) \varphi(\epsilon) \, d\epsilon, \quad \text{with} \quad L_0(u) = L(u) \]

Show that \( \bar{x} \) is the root of
\[ c(1 - \alpha) + \alpha^\lambda \int_0^\infty ... \int_0^\infty L'(\bar{x} - \epsilon_1 - \epsilon_2 - ... - \epsilon_\lambda) \varphi(\epsilon_1) \varphi(\epsilon_2) ... \varphi(\epsilon_\lambda) \, d\epsilon_1 ... \, d\epsilon_\lambda = 0 \]
or equivalently of
\[ c(1 - \alpha) + \alpha^\lambda \int_0^\infty L'_{\lambda-1}(\bar{x} - \epsilon) \varphi(\epsilon) \, d\epsilon = 0 \]
36. For a hydroelectric system, we define the following terms:

\[ z_i = \text{demand for electric energy during } i\text{th interval} \]
\[ r_i = \text{water flowing into reservoir during } i\text{th interval, random variable} \]
\[ q_i = \text{density distribution of } r_i \]
\[ K = \text{maximum amount of supplementary electrical energy supplied by nonhydroelectric means (steam generated power)} \]
\[ c = \text{cost of supplementary electrical energy} \]
\[ p = \text{penalty cost if demand exceeds supply} \]
\[ \alpha = \text{discount factor} \]
\[ R_i = \text{volume of water in reservoir at beginning of } i\text{th interval} \]
\[ u_i = \text{volume of water used during } i\text{th interval} \]

Show that the minimum cost over \( u_i \) during \( i\)th interval is

\[
c \cdot \min (K, z_i - u_i) + p \cdot \max (0, z_i - u_i - K) = \begin{cases} 
  cK + p(z_i - u_i - K), & \text{if } z_i - K \geq 0 \\
  c(z_i - u_i), & \text{if } 0 \leq u_i \leq z_i - K \\
  \max (0, z_i - K) \leq u_i \leq z_i 
\end{cases}
\]

Graph the penalty as a function of \( u_i \). Show that the functional equation for an \( i\)-stage process for minimum cost is

\[
f_i(R_i) = \min_{0 \leq u_i \leq \min(z_i, R_i)} \left[ c \cdot \min (K, z_i - u_i) + p \cdot \max (0, z_i - u_i - K) + \alpha \int_0^\infty f_{i-1}(R_i + r_i - u_i) \cdot q_i(r_i) \, dr \right]
\]

where

\[
R_i = R_{i+1} + r_{i+1} - u_{i+1} \quad (i = 1, 2, ..., T - 1)
\]

Since \( u_1\) = \( \min(z_1, R_1) \), show for the one-stage process that

\[
f_1(R_1) = \begin{cases} 
  cK + p(z_1 - R_1 - K), & \text{if } z_1 - K > 0 \\
  c(z_1 - R), & \text{if } 0 \leq R_1 \leq z_1 - K \\
  0, & \text{if } z_1 \leq R_1 
\end{cases}
\]

(Arrow, K. J., Karlin, S., and Scarf, H. [2].)

37. The Arrow-Harris-Marschak equation allowing for negative stocks takes the following form:

\[
\lambda(x) = \min_x \left[ k(z) + l(x + z) + a \int_0^\infty \lambda(x + z - \epsilon) \, d\varphi(\epsilon) \right]
\] (1)
where

\[ l(y) = h(y) + p \int_{y}^{\infty} (\epsilon - y) \, dp(\epsilon) \]

is expected loss during a period for which stock level after ordering is \( y \).

Define

\[
\begin{align*}
I(y) &= h(y) + p \int_{y}^{\infty} (\epsilon - y) \, dp(\epsilon) \\
\varphi(\epsilon) &= e^{-\epsilon} = \text{exponential density function} \\

h &= \text{storage cost per commodity per period} \\
p &= \text{penalty per unit of storage} \\

k &= \text{cost of placing an order}
\end{align*}
\]

Show that the optimal cost function is of the \((s, S)\) type where \( s \) is the reorder point and \( S \) is the optimal starting stock

\[
\lambda(x) = \begin{cases} 
  l(x) + a \int_{0}^{x} \lambda(x - \epsilon) \, dp(\epsilon), & x \geq s \\
  k + l(S) + a \int_{0}^{x} \lambda(S - \epsilon) \, dp(\epsilon), & x < s
\end{cases}
\]

Show that for \( x \geq s \)

\[
\lambda(x) = l(x) + a \int_{0}^{x-s} \lambda(x - \epsilon) \, dp(\epsilon) + a \lambda(s) \left[ 1 - \varphi(x - s) \right] 
\]

and that for \( x = s \)

\[
\lambda(s) = \frac{l(s)}{1 - a}
\]

(Arrow, K. J., Karlin, S., and Scarf, H. [2].)

38. Referring to the previous problem, we have for \( x \geq s \)

\[
\lambda(x) = l(x) + a \int_{0}^{x-s} \lambda(x - \epsilon) \, dp(\epsilon) + a \lambda(s) \left[ 1 - \varphi(x - s) \right] \\
\varphi(\epsilon) = e^{-\epsilon} \\
l(y) = hy + pe^{-y}
\]

Let \( t = x - \epsilon \) and \( u(x) = \int_{0}^{x} \lambda(t) \, e^{t} \, dt \). Show that

\[
\lambda(x) = hx + pe^{-x} + \frac{a}{1 - a} \left[ hse^{s} + p \right] e^{-x} + a \int_{s}^{x} \lambda(t) \, e^{t-x} \, dt
\]
Using the boundary condition \( u(s) = 0 \), show that

\[
u'(x) = au(x) + hxe^x + p + \frac{a}{1-a} \left( hse^s + p \right)\]

Show that the solution is

\[
u(x) = e^{ax} \int_s^x e^{-at} \left( \frac{a}{1-a} hse^s + \frac{p}{1-a} + hte^t \right) dt\]

or

\[
\lambda(x) = u'(x) e^{-x} \\\lambda(x) = \frac{hx}{1-a} - \frac{ah}{(1-a)^2} + e^{(a-1)(x-s)} \left[ \frac{ah}{(1-a)^2} + \frac{pe^{-s}}{1-a} \right]
\]

(Arrow, K. J., Karlin, S., and Scarf, H. [2].)

39. Referring to the previous problem and using the fact that \( S \) is the optimal stock level, it follows that

\[
\lambda'(S) = 0, \quad \lambda(S) - \lambda(S) = k
\]

\[
\lambda'(x) = \frac{h}{1-a} + (a-1) e^{(a-1)(x-s)} \left[ \frac{ah}{(1-a)^2} + \frac{pe^{-s}}{1-a} \right]
\]

Show that

\[
S - s = \frac{1}{1-a} \log q; \quad s = \log \frac{p(1-a)}{h(q-a)}
\]

where

\[
q = 1 + \frac{(1-a)^2}{h} k + \log q; \quad q = a + (1-a) \frac{p}{h} e^{-s}
\]

Plot \( D = (S - s) \) as function of \( q \).

When \((k/h)(1-a)^2 < < 1\), \( q \approx 1 \), and taking \( \log q = t - (t^2/2) \),

where \( q = 1 + t \) show that

\[
q \approx 1 + (1-a) \sqrt{\frac{2k}{h}}; \quad D \approx \sqrt{\frac{2k}{h}} - (1-a) \frac{k}{h}
\]

(Arrow, K. J., Karlin, S., and Scarf, H. [2].)

40. We are given an inventory and production problem in which it is desired to minimize over the time interval \((0, T)\) the cost of production and holding inventory subject to a known demand. Define
8. INVENTORY PROBLEMS

\[ z(t) = \text{the rate of production at time } t \]
\[ r(t) = \text{the product demand at time } t \]
\[ y(t) = \text{the number of units in stock at time } t \]
\[ h = \text{the holding cost of inventory per unit, a constant} \]
\[ c(z) = \text{cost of production, a monotone increasing convex function} \]
\[ s(t) = \int_0^t r(\tau) \, d\tau - y(0) \]

We desire to minimize over \( z \)

\[
J(z) = \int_0^T \left\{ c[z(t)] + h \left[ y(0) + \int_0^t z(\tau) \, d\tau - \int_0^t r(\tau) \, d\tau \right] \right\} \, dt
\]

Note that a material balance satisfies

\[
y(t) = y(0) + \int_0^t z(\tau) \, d\tau - \int_0^t r(\tau) \, dt
\]

and that \( z(t) \) is restricted by

\[
y(0) + \int_0^t z(\tau) \, d\tau \geq \int_0^t r(\tau) \, d\tau
\]

with \( z(t) \geq 0 \). Arrow, Karlin, and Scarf have deduced that the solution to the problem has the following form, if \( c(z) \) is linear:

\[
z_0(t) = 0 \quad \text{until} \quad s(t) = y(0), \quad \text{after which} \quad z_0(t) = r(t).
\]

Deduce this from a dynamic programming formulation.

(Arrow, K. J., Karlin, S., and Scarf, H. ([2].))
1. Introduction

Practically every problem presented in this book can be reformulated in probabilistic terms. To illustrate how stochastic processes can be handled by dynamic programming, some typical problems will be posed in stochastic garb. That a common approach suffices for both deterministic and stochastic processes will be disclosed via the mechanism of dynamic programming. All that is required is suitable redefining of the state variables consistent with the concept of expectation.

In this chapter, in order to compare deterministic, stochastic, and adaptive processes, we will set up each in a general but somewhat abstract manner. In this way, we can observe the anatomy of the transformations from stage to stage of the state variables and the sequence of control vectors that yield the optimal policy. Having exposed the essentials of these processes, we illustrate the stochastic and adaptive processes by means of problems of allocation, averaging control, and catalyst replacement.

A general formulation of deterministic processes is presented in Section 2. This is illustrated by generalized allocation problem. In a similar manner, in Section 3 a general formulation of stochastic processes is set forth. A generalized stochastic allocation problem employing the concept of expected return illustrates the point. A comparison of stochastic and deterministic processes is drawn in Section 4. Identification of stochastic elements in many processes and in particular, chemical
processes, is made. Section 5 deals with a stochastic version of an allocation problem previously discussed. A stochastic catalyst regeneration model is described in Section 6. A version of an averaging control problem is treated as a stochastic problem due to a random variable in Van der Pol's equation. In view of the manifold ways probabilistic elements may enter a process, Section 8 sets forth some of these possibilities. How stochastic matrices appear in chemical engineering problems is the subject of Section 9. The stochastic version of the tankage problem of Chapter 7 appears in Section 10 in terms of conditional probabilities. The last three sections deal with a special stochastic model, the adaptive model. Section 11 is devoted to its general formulation, while Section 12 considers an averaging adaptive control model, and Section 13 discusses an adaptive catalyst replacement model.

2. Deterministic Processes. General Formulation

Dynamic programming offers a uniform way of solving deterministic, stochastic, and adaptive problems. The special appeal of dynamic programming is this universality. In order to spell out the relationship between deterministic and stochastic problems, let us first set up a general but somewhat abstract formulation of these two processes.

For the deterministic model, let us consider an $N$-stage process, where the state of the system is given by $s_i$, a vector of one or more elements. The transformation of the state vector from stage to stage is executed by manipulating the control vector $v_i$, $i = 1, 2, ..., N$. The transformation $T_i(s_i, v_i)$ is a function of the state vector and the control vector at each stage. If we count the stages backward so that stage $N$ means $N$ stages remaining, the set of transformations beginning with the initial stage $N$ and ending with the final stage 1 appears as follows:

$$
\begin{align*}
    s_{N-1} &= T_N(s_N, v_N) \\
    s_{N-2} &= T_{N-1}(s_{N-1}, v_{N-1}) \\
    \vdots & \quad \vdots \\
    s_0 &= T_1(s_1, v_1)
\end{align*}
$$

(1)

Substituting the equations of Eqs. (1) into one another in order, we have the final state $s_0$ expressed in terms of the initial state $s_N$:

$$
s_0 = T_1(T_2(T_3(\ldots T_N(s_N, v_N), v_{N-1})) \ldots)
$$

(2)
Since the transformations $T_i$ are generally identical, Eq. (2) may be written more compactly as

$$s_0 = T^N(s_N, v_N)$$

where the exponent $N$ means successive applications of the transformation.

As a consequence of the transformations in Eq. (1), there is a set of corresponding decision vectors $(v_N, v_{N-1}, \ldots, v_1)$ called a policy. When the transformations are made to satisfy criterion of performance, the set of optimizing decision vectors is called the optimal policy.

Let us now consider maximizing the cumulative return over an $N$-stage process:

$$P = \sum_{i=1}^{N} g_i(s_i, v_i)$$

where

$$g_i(s_i, v_i) = \text{the return from the } i\text{th stage, which is a function of the state vector } s_i, \text{ and the control vector } v_i$$

$$P = \text{the cumulative return}$$

We define

$$f_N(s_N) = \text{the maximum return over the } N \text{ remaining stages of the process beginning in state } s_N \text{ and using an optimal policy, } N = 1, 2, \ldots$$

$$f_N(s_N) = \text{Max}_{v_i} \sum_{i=1}^{N} g_i(s_i, v_i)$$

Expanding Eq. (8) and using the Principle of Optimality, we find that

$$f_N(s_N) = \text{Max}_{v_N} [g_N(s_N, v_N) + f_{N-1}(s_{N-1})]$$

By virtue of Eq. (1), this may be written as

$$f_N(s_N) = \text{Max}_{v_N} [g_N(s_N, v_N) + f_{N-1}(T(s_N, v_N))]$$

$$f_1(s_1) = \text{Max}_{v_1} [g_1(s_1, v_1)]$$

The path from Eq. (1) through Eq. (11) has been taken many times throughout this book.

For the stochastic model, the transformation from stage \( i \) to stage \((i - 1)\) is fraught with uncertainty. As a consequence of the transformation \( T_i(s_i, v_i) \) the known state vector \( s_i \) is cast into a probabilistic state \( z_{i-1} \) with an associated distribution function \( dG(s_i, z_{i-1}, v_i) \). The distribution function depends on the known state \( s_i \), the stochastic state \( z_{i-1} \), and the control vector \( v_i \).

In this process, we start with a known state \( s_i \), which upon transformation yields a stochastic vector \( z_{i-1} \). Before a decision is made at stage \((i - 1)\), we assume that the actual state \( s_{i-1} \) is observed and is known. In other words, among the set of \textit{a priori} stochastic vectors \( z_{i-1} \) exists \( s_{i-1} \), the actual state vector.

The set of transformation in the stochastic case appears as

\[
\begin{align*}
z_{N-1} &= T_N(s_N, v_N) \\
z_{N-2} &= T_{N-1}(s_{N-1}, v_{N-1}) \\
& \vdots \\
z_0 &= T_1(s_1, v_1)
\end{align*}
\]

In the stochastic process, we cannot back substitute, as we did in Section 2, to express the final state as a function of the initial state. This is due to the fact that we do not know the results of a transformation until an observation is made.

Since the \( z_i \) terms are probabilistic, the \( v_i \) vectors are also stochastic in the sense that the application of the \( v_i \) yields an uncertain result. Considering once again the maximization of

\[
P = \sum_{i=1}^{N} g(s_i, v_i)
\]

as our objective, it is obvious that the return function itself is stochastic in view of uncertainty in \( z_i \) and \( v_i \). For this reason, we can no longer talk about maximizing the return.

We must compromise our objectives and settle for a goal that maximizes in the face of these uncertainties. The utilization of the statistical concept of expectation meets these needs satisfactorily. We revise our objective to maximize the \textit{expected return}.

Thus, we define

\[
F_N(s_N) = \text{the maximum expected value of the return function over} \\
\text{over the } N \text{ remaining stages starting in state } s_N \\
\text{and using an optimal policy}
\]
The profitability expression is given by

\[ F_N(s_N) = \max_{\nu_N} E \left[ \sum_{i=1}^{N} g_i(T_i(s_i, \nu_i), \nu_i) \right] = \max_{\nu_N} E \left[ \sum_{i=1}^{N} g_i(z_{i-1}, \nu_i) \right] \tag{4} \]

where \( E \) stands for the expectation.†

Combining the linearity of the expectation and the Principle of Optimality, we can write

\[ F_N(s_N) = \max_{\nu_N} E \left[ g_N(z_{N-1}, \nu_N) + f_{N-1}(z_{N-1}) \right] \tag{5} \]

\[ F_N(s_N) = \max_{\nu_N} \left[ \int_{-\infty}^{\infty} [g_N(z_{N-1}, \nu_N) + f_{N-1}(z_{N-1})] dG(s_N, z_{N-1}, \nu_N) \right] \tag{6} \]

\[ F_1(s_1) = \max_{\nu_1} \left[ \int_{-\infty}^{s_1} g_1(z_0, \nu_1) dG(s_1, z_0, \nu_1) \right] \tag{7} \]

In the examples to be discussed here, we will consider discrete density distribution functions where the distribution \( dG(s_N, z_{N-1}, \nu_N) \) will be replaced by \( \{\pi_i\} \). For the discrete case, the return function is

\[ F_N(s_N) = \max_{\nu_N} \sum_{j=1}^{M} \left[ g_N(z_{N-1}, \nu_N) + f_{N-1}(z_{N-1}) \right] \pi_j \tag{8} \]

† By definition, the expectation of a random variable \( x \) whose probability density function is \( g(x) \) is, for the continuous case,

\[ E(x) = \int_{-\infty}^{\infty} x g(x) \, dx \]

For the discrete density distribution, the expectation is

\[ E(x) = \sum_{i=1}^{N} x_i \pi_i \]

Using the Riemann-Stieltjes integral, one expression can be used to describe both the continuous and discrete expectations:

\[ E(x) = \int_{-\infty}^{\infty} x \, dG(x) \]

An important property of the expectation is its linearity

\[ E(x + y) = E(x) + E(y) \]

It is this property that we exploit in order to apply the Principle of Optimality.
where

\[ 0 \leq p_j \leq 1, \quad j = 1, 2, \ldots, M \]  

(9)

\[ \sum_{j=1}^{M} p_j = 1 \]  

(10)

The probability density function is defined over the \( M \) possible discrete states that stochastic vector \( z_{N-1} \) may assume.

4. Stochastic versus Deterministic Process

All mathematical models are fictions, some more believable than others. The word model itself implies a copying or a reconstruction of something more real or perhaps more valuable than the model itself. In deterministic models, the cause and effect relationships are unequivocally set forth. Bold statements (bold mathematical statements, that is) can be made about how the process proceeds from stage to stage. The sequence of decision variables and the corresponding return functions are known with absolute mathematical certainty. The advantage of the deterministic model is that it concentrates on the principle aspects of the problem. As a result, this model often yields equations simple enough to be handled mathematically.

Stochastic models attempt to weave into the mathematical fabric more threads of reality. Rather than attempt to examine every nuance of uncertainty, stochastic models smooth out the probabilistic behavior by using some kind of an averaging procedure. Typical of these is expectation. What certainty is to the deterministic model, expectation is to the stochastic model. Each provides a mechanism for describing the transformation from stage to stage.

Stochastic problems appear for many reasons. Some of these are:

1. The equations describing the process are only approximations, since the "true" equations are not known. The transformations from stage to stage are probabilistic.
2. The state of the system cannot be clearly identified.
3. The physical, chemical, and economic data in the models are estimates of varying degrees of certainty.
4. The stochastic variables cannot be clearly identified.
5. The probability density function is not known.
6. The probability density function is known, but the parameters, such as \( \mu \) and \( \sigma \), vary.
(7) The objectives of the problem change from time to time.
(8) A great deal of uncertainty exists in the measurements taken in the process.

In particular, in chemical processes stochastic problems exist among others due to:

(1) Uncertainties in flow measurements, temperature measurements, etc.
(2) Uncertainties in sampling procedure and the execution of the sampling.
(3) Uncertainties in the sample composition itself.
(4) Uncertainties in the sample size.
(5) Uncertainties in the laboratory measurements.
(6) Uncertainties in the laboratory personnel skills.
(7) Uncertainties in the relative frequency of sampling.
(8) Paucity of data that goes into model.
(9) Extrapolation of physical and chemical data.
(10) Infrequent calibration of meters and instruments.

A comparison of Sections 2 and 3 shows that dynamic programming provides a common means for formulating deterministic and stochastic problems. Provided the probability density function is known or can be assumed and provided the return function is defined in terms of expected value, the computational procedure for both cases is identical using dynamic programming.

It has been pointed out in Chapter 5, Section 10 that deterministic processes may be solved by the differential calculus as well as by dynamic programming. The differential calculus yields at one fell swoop the entire set of \( v_i \) terms as the solution of a set of \( N \) equations. The dynamic programming approach, on the other hand, yields the same results by developing a sequence of return functions and manipulated variables \( v_i \), one by one over the \( N \) stages. Both the differential calculus and dynamic programming can be used in deterministic processes.

In stochastic processes the results of each transformation are not known for certain when the decision is made. Before the next decision is made, however, the state is considered to be known due to observation. This means that stochastic processes, when handled by dynamic programming, are feedback processes. The observed current state of the system is required in order to make the next decision. As a consequence for the stochastic process, it is not possible to determine at the beginning of the process the values of all the decision variables over all the stages, as is done in effect by the differential calculus solution.
5. Stochastic Allocation Problem

In Chapter 5, Section 15, an allocation problem was set forth in its deterministic form. It is our intention to present the same problem as a stochastic problem.

The deterministic equations are

\[ f_N(x) = \max_{0 < y < x} \left[ g(y) + h(x - y) + f_{N-1}(ay + b(x - y)) \right] \]  

\[ f_i(x) = \max_{0 < y < x} \left[ g(y) + h(x - y) \right] \]  

with

\[ 0 \leq a, \ b \leq 1 \]  

Let us assume in the stochastic process that with the choice of \( y \) that \( g(y) \) may take one of two values. With a probability \( p_1 \), it takes the value \( g_1(y) \) with \( y \) decreasing to \( a_1y \), and with probability \( p_2 = (1 - p_1) \), it takes the value of \( g_2(y) \) with \( y \) decreasing to \( a_2y \). Two possibilities exist for the return function \( h(x - y) \). With probability \( q_1 \), it assumes the value \( h_1(x - y) \) with \( (x - y) \) decreasing to \( b_1(x - y) \). With probability \( q_2 = (1 - q_1) \), it assumes the value of \( h_2(x - y) \) with \( (x - y) \) decreasing to \( b_2(x - y) \). Since the events described by \( p_1, p_2, q_1, \) and \( q_2 \) are independent, the \( dG \) function of Eqs. (6) and (7) of Section 3 may be described with its corresponding random variables as

\[ p_1q_1, \ a_1y + b_1(x - y); \quad p_2q_1, \ a_2y + b_1(x - y) \]  

\[ p_1q_2, \ a_1y + b_2(x - y); \quad p_2q_2, \ a_2y + b_2(x - y) \]  

If we let

\[ P_1 = p_1q_1; \quad P_3 = p_2q_1 \]  

\[ P_2 = p_1q_2; \quad P_4 = p_2q_2 \]  

then we may observe that

\[ \sum_{i=1}^{4} P_i = 1 \]  

\[ 0 \leq P_i \leq 1 \]  

Thus, the distribution, Eq. (5), satisfies the definition of a discrete probability density function.

We now define:

\[ F_N(x) = \text{the maximum expected return over the } N \text{ remaining stages} \]  

beginning in state \( x \) and using an optimal policy
6. STOCHASTIC CATALYST REGENERATION MODEL

Using the definition of $F_N(x)$ and Eqs. (1)-(7), the stochastic allocation optimization may be written as

$$F_N(x) = \max_{0 \leq y \leq x} \{ p_1 q_1 (g_1(y) + h_1(x - y)) + f_{N-1}(a_1 y + b_1(x - y)) + p_2 q_2 (g_2(y) + h_2(x - y)) + f_{N-1}(a_2 y + b_2(x - y)) \}$$

(8)

$$F(x) = \max_{0 \leq y \leq x} \{ p_1 q_1 (g_1(y) + h_1(x - y)) + p_2 q_2 (g_2(y) + h_2(x - y)) \}$$

(9)

where

$$0 \leq a_1, a_2, b_1, b_2 \leq 1$$

(10)

$$0 \leq p_1, p_2, q_1, q_2 \leq 1$$

(11)

$$\begin{align*}
p_1 + p_2 &= 1 \\
q_1 + q_2 &= 1
\end{align*}$$

(12)

In this formulation, we see that the original state variable $x$ is transformed into a stochastic variable with four different possibilities

$$a_1 y + b_1(x - y); \quad a_1 y + b_2(x - y)$$

$$a_2 y + b_1(x - y); \quad a_2 y + b_2(x - y)$$

These correspond to $(z_{N-1})$ in Eq. (1) of Section 3. Equations (8) and (9) correspond directly to Eq. (8) of Section 3.

If the $p_1, p_2, q_1, q_2$ are known, the calculation of the return function is exactly the same as for the deterministic case. The quantity of calculations is considerably greater, however, at each stage, since the four possibilities must be evaluated.

6. Stochastic Catalyst Regeneration Model

In Chapter 2, Section 5, a simple catalyst replacement or regeneration process is described. To introduce probabilistic concepts into the catalyst replacement or regeneration problem, we will reformulate the simplest model, Model I, in stochastic terms. In the stochastic version of Model I, we do not know for sure, prior to regeneration, the outcome of the regeneration process. All we know is that a "good" catalyst will be produced with a probability $p$, and a "poor" catalyst
with probability \( q = 1 - p \). In this stochastic model, a "good" reactor remains "good" for the entire operating cycle and a "poor" reactor remains "poor" for the entire operating cycle.

In the Model I, the profit-time function is known. In the stochastic version of Model I, we have profit-time functions for both a "good" and a "poor" catalyst.

If the catalyst responds well to regeneration and then acts as a "good" catalyst, we define the following sequence of profit terms:

\[
P_0 = \text{the daily profit from a "good" reactor, zero days old} \\
P_1 = \text{the daily profit from a "good" reactor, one day old} \\
P_2 = \text{the daily profit from a "good" reactor, two days old} \\
\vdots \\
P_N = \text{the daily profit from a "good" reactor, } N \text{ days old}
\]

If the catalyst does not respond well to regeneration and then acts as a "poor" catalyst, we define the following sequence of profit terms:

\[
Q_0 = \text{the daily profit from a "poor" reactor, zero days old} \\
Q_1 = \text{the daily profit from a "poor" reactor, one day old} \\
Q_2 = \text{the daily profit from a "poor" reactor, two days old} \\
\vdots \\
Q_N = \text{the daily profit from a "poor" reactor, } N \text{ days old}
\]

We note that

\[ P_i > Q_i, \quad i = 0, 1, 2, ..., N \]

We now define the following terms:

\[
f_N(t) = \text{the expected return from an } N \text{-stage process beginning with a "good" catalyst that is } t \text{ days old and using an optimal policy; } N = 1, 2, ... \]

\[ g_N(t) = \text{the expected return from an } N \text{-stage process beginning with a "poor" catalyst that is } t \text{ days old and using an optimal policy; } N = 1, 2, ... \]

\[ R = \text{the cost of regenerating the catalyst.} \]

We can now formulate the functional equations:

\[
f_N(t) = \max \{ P_t + f_{N-1}(t + 1), \quad \}
\]

\[
f_1(t) = P_1 \\
g_N(t) = \max \{ Q_t + g_{N-1}(t + 1), \quad \}
\]

\[
g_1(t) = Q_1 \quad 0 \leq p, \quad q \leq 1
\]
7. STOCHASTIC AVERAGING CONTROL PROBLEM

Equations (7) and (8) refer to the “good” reactor, while Eqs. (9) and (10) refer to the “poor” reactor. In Eq. (7), the top line evaluates the expected return for the possibility of continuing to operate the reactor. The bottom line evaluates the expected return for the possibility of shutting down for one day and regenerating the catalyst. For the top line of Eq. (7), the term $P_t$ is the profit from a “good” reactor “$t$” days old and the term $f_{N-1}(t + 1)$ is the expected return from the remaining $(N - 1)$ stages beginning with a “good” catalyst $(t + 1)$ days old. The stochastic elements enter in the bottom line of Eq. (7) where we do not know with certainty the outcome of the regeneration process, but only the expected outcome. The first term “$R$” is the cost of the regeneration. The term $pf_{N-1}(0)$ is the expected return for the probability of the regeneration yielding a “good” catalyst for the remaining $(N - 1)$ stages starting with a catalyst zero days old. The last term $(1 - p)g_{N-1}(0)$ is the expected return for the probability of the regeneration yielding a “poor” catalyst for remaining $(N - 1)$ stages starting with a catalyst zero days old. Equation (7) states we must choose the maximum of the two contending possibilities.

Equation (9) can be discussed in a manner parallel to that of Eq. (7). To solve these functional equations both $f_N(t)$ and $g_N(t)$ must be calculated.

The stochastic model discussed here is a very simple case of a “black” or “white” situation. In actual practice there are a variety of stochastic problems that range from this simple model to self-adaptive models. In any case, the principles illustrated here can be extended to more complicated problems.

7. Stochastic Averaging Control Problem

Let us consider a stochastic version of the averaging control problem given in Chapter 6, Section 4. We desire to minimize $J(w)$ over $w$:

$$J(w) = \int_0^T [x^2 + (x')^2 + aw^2] \, dt \quad (1)$$

subject to

$$x'' + \mu(x^2 - 1) x' + x = w(t) + r(t) \quad (2)$$

$$x(0) = c_1 \quad (3)$$

$$x'(0) = c_2 \quad (4)$$

This problem is characterized by the presence of a random variable
r(t), as an additive term. The function r(t) has a distribution function dG(r). We must minimize now in the face of this random disturbance.

Using a discrete formulation, the equations become

\[ J_N(w_k) = \sum_{k=1}^{N} (x_k^2 + y_k^2 + aw_k^2) \Delta \]  
(5)

\[ x_{k+1} = x_k + y_k \Delta \]  
(6)

\[ y_{k+1} = y_k + \left[ w_k + r_k - \mu(x_k^2 - 1) y_k - x_k \right] \Delta \]  
(7)

\[ x_N = c_1 \]  
(8)

\[ y_N = c_2 \]  
(9)

The control vector is furthermore constrained by

\[ w_{\text{Min}} \leq w_k \leq w_{\text{Max}} \]  
(10)

For simplicity sake, let us assume that \( r_k \) may take only two values. With probability \( p \), \( r_k = +R \) and with probability \( q = 1 - p \), \( r_k = -R \).

We now define

\[ F_N(c_1, c_2) = \min_{w_k, r_k} E J_N(w_k) = \text{the minimum over } w_k \text{ of the expectation over } r_k \text{ of the return } J_N(w_k) \text{ over the } N \text{ remaining stages of the process starting in the state } (c_1, c_2), \text{subject to Eqs. (6)-(10), and using an optimal policy} \]

\[ F_N(c_1, c_2) = \min_{w_N} \left[ p \left( (c_1^2 + c_2^2 + aw_N^2) \Delta + F_{N-1}(c_1 + c_2 \Delta, c_2 \right. \\
\left. + [w_N + R - \mu(c_1^2 - 1) c_2 - c_1] \Delta) \right), \right. \]
\[ + q \left( (c_1^2 + c_2^2 + aw_N^2) \Delta + F_{N-1}(c_1 + c_2 \Delta, c_2 \right. \\
\left. + [w_N - R - \mu(c_1^2 - 1) c_2 - c_1] \Delta) \right) \]  
(12)

Since \((q + p) = 1\), Eq. (12) reduces to

\[ F_N(c_1, c_2) = \min_{w_N} \left[ (c_1^2 + c_2^2 + aw_N^2) \Delta + pF_{N-1}(c_1 + c_2 \Delta, c_2 \right. \\
\left. + [w_N + R - \mu(c_1^2 - 1) c_2 - c_1] \Delta \right) \]
\[ + qF_{N-1}(c_1 + c_2 \Delta, c_2 + [w_N - R - \mu(c_1^2 - 1) c_2 - c_1] \Delta) \]  
(13)

\[ F_1(c_1, c_2) = \min_{w_1} \left[ c_1^2 + c_2^2 + aw_1^2 \right] \Delta \]  
(14)

The formulation for the stochastic case is essentially identical with the deterministic case given by Eqs. (11) and (12) in Chapter 6, Section 4.
The distinguishing characteristic of the stochastic version is the presence of $pF_{N-1}()$, and $qF_{N-1}()$ describing the two stochastic possibilities.

8. Variations on a Theme

In the last several sections, we took the probability density function to be a discrete Bernoulli distribution. One event occurred with a probability $p$, the other possible event occurred with probability $q = (1 - p)$. The probability density function was common for all stages of the process. We may enlarge upon this idea of the Bernoulli distribution and consider a number of variations. Referring specifically to the Section 7, we list several of the many possibilities.

1. The probabilities $p$, $q$, are a function of the stage considered. There is a set of probabilities: $p_N$, $q_N$ at stage $N$, $p_{N-1}$, $q_{N-1}$, at stage $N - 1$, etc.

2. The probabilities are a function of the decision at each stage:

   \[ p_N = p_N(w_N), \quad q_N = q_N(w_N), \]
   \[ p_{N-1} = p_{N-1}(w_{N-1}), \quad q_{N-1} = q_{N-1}(w_{N-1}), \quad \text{etc.} \]

3. The probabilities are a function of the initial state, that is,

   \[ p_N = p_N(c_1, c_2), \quad q_N = q_N(c_1, c_2) \]

4. Various combination of items (1), (2), and (3) such as the probabilities are a function of the stages remaining, the decision made, and the initial state.

5. One other important stochastic aspect is that we may recognize is the functional dependence of the probabilities on the stage, on the decision, or on the initial state without knowing the explicit relationship. This apparent great difficulty may be coped with by exploiting an adaptive model in which the process itself is used to elucidate these relationships.

To illustrate the first possibility, if $p_i$ and $q_i$ are known over the $i$ stages of the process, Eqs. (13) and (14) of Section 7 may be written as

\[
F_N(c_1, c_2) = \text{Min}_{w_N} \left[ (c_1^2 + c_2^2 + aw_N^2) \Delta \right. \\
+ p_N F_{N-1}(c_1 + c_2 \Delta, c_2 + [w_N + R - \mu(c_1^2 - 1) c_2 - c_1] \Delta) \\
+ q_N F_{N-1}(c_1 + c_2 \Delta, c_2 + [w_N - R - \mu(c_1^2 - 1) c_2 - c_1] \Delta) \right] \quad (1)
\]

\[
F_i(c_1, c_2) = \text{Min}_{w_i} \left[ c_1^2 + c_2^2 + aw_i^2 \right] \Delta \quad (2)
\]

\[
p_i + q_i = 1, \quad i = 1, 2, ..., N \quad (3)
\]
9. Stochastic Matrix in Chemical Engineering

The averaging control problem with stochastic properties is typical of a large class of stochastic problems where the control equation appears in matrix form as

\[
\frac{dx}{dt} = Ax + r(t)
\]  

(1)

where some function is to be maximized or minimized.

If the \( r(t) \) can be identified or if some information is available on its probability density function, then the methods presented are useful.

Oftentimes, however, one does not know enough to state that the above equation is a valid stochastic expression. The probability effect may not be a simple additive term. In these cases, all we can say is that theory tells us that

\[
\frac{dx}{dt} = Ax
\]  

(2)

In practice, however, the right-hand side does not equal the left-hand side all of the time. To circumvent this difficulty, the elements of the matrix \( A \) must be evaluated periodically in order to have a working relationship. Let us consider Eq. (2) as describing a chemical reaction

\[
X_1 \xrightarrow{k_1} X_2 \xrightarrow{k_2} X_3
\]  

(3)

\[
\frac{dx_1}{dt} = -k_1x_1 + k_2x_2
\]

\[
\frac{dx_2}{dt} = +k_1x_1 - (k_2 + k_3)x_2
\]  

(4)

\[
\frac{dx_3}{dt} = +k_2x_2
\]

where

\[
k_i = A_i \exp \left( -\frac{E_i}{RT} \right)
\]  

(5)

It is quite easy to see that with the uncertainties in measurements variability in catalyst activity, the formation of side reactions and by products that the elements of the matrix \( A \) of Eq. (4).

\[
A = \begin{pmatrix}
-k_1 & k_2 & 0 \\
k_1 & - (k_2 + k_3) & 0 \\
0 & k_2 & 0
\end{pmatrix}
\]

are probabilistic. Mathematicians refer to \( A \) as a stochastic matrix.

In some stochastic problems the past history of the process determines the probability density functions for the succeeding periods. This concept can be handled conveniently through conditional probabilities. To illustrate this, let us turn to a stochastic version of the single tank control problem in Chapter 7, Section 2.

In the stochastic case, there are a number of probabilistic aspects. First, we possess from a forecast or prognostication an estimate of the desired schedule which we call $\dot{Q}_E^*$. The effluent flow rate is also a stochastic variable. Equation (3) of Chapter 7, Section 2 is now written in terms of the estimated effluent $(\dot{Q}_E)_i$ as

$$\dot{Q}_E)_i = K_E \sqrt{h_i}$$ (1)

The error squared between the desired flow $Q_E^*$ and the delivered flow $Q_E$, $(Q_E^* - Q_E)^2$, may be due to the uncertainties in $(\dot{Q}_E^*)_i$, or $(Q_E)_i$, due to uncertainties in the orifice coefficients and leakage. Neither the desired flow rate $(Q_E^*)_i$ nor the actual effluent $(Q_E)_i$ is known for certain until the $i$th stage has passed. For these reasons, it is not reasonable to require that the error squared be minimized as with the deterministic case. We must compromise and settle for a less exact measure of performance. In this stochastic model, the expected value of the error squared over $N$ stages of time is a convenient measure.

In this model, we consider that for each stage of the $N$-stage process that a record has been kept of the $\dot{Q}_E^*, Q_E^*, \dot{Q}_E, Q_E, h, h'$. As a consequence of this record over $M$ cycles of the $N$-stage process, a probability density function for the error squared for each stage of the $N$-stage process has been developed. In general, the probability distribution function for each stage is unique. Furthermore, the probability distribution functions are considered not to vary with time.

The frequency distribution for each stage based on the completion of $M$ cycles is expressed as the following conditional frequency:

$$p[(Q_E^* - Q_E)^2 | \dot{Q}_E^*, h, h'] = \frac{p[(Q_E^* - Q_E)^2, \dot{Q}_E^*, h, h']} {p[\dot{Q}_E^*, h, h']}$$ (2)

This conditional frequency determines the distribution of the error squared given values for $\dot{Q}_E^*, h, h'$. The quantity $h$ is the head in the tank at the beginning of the current stage, and $h'$ is the head at the beginning of the next stage. The function $p[(Q_E^* - Q_E)^2, \dot{Q}_E^*, h, h']$ is the joint probability distribution function of $(Q_E^* - Q_E)^2, \dot{Q}_E^*, h, h'$. 

Similarly, \( p[\hat{Q}_E^*, h, h'] \) is the joint probability distribution function of \( \hat{Q}_E^*, h, h' \).

We recall that the conditional expectation of the random variable \( Y \) given the random variable \( X \) is

\[
E[Y \mid X] = \frac{\sum_k y_k p(x_j, y_k)}{p(x_j)}
\]

(3)

where \( p(x_j, y_k) \) is the joint probability distribution function of \( X \) and \( Y \) and \( p(x_j) \) is the probability distribution function for \( X \).

Since the expectation of a sum is equal to the sum of the expectations, we may write

\[
E\left( \sum_{i=1}^{N} (Y_i \mid X_i) \right) = E(Y_1 \mid X_1) + E(Y_2 \mid X_2) + \ldots + E(Y_N \mid X_N)
\]

(4)

If we associate \( Y_i \) with \( (Q^*_E - Q_i) \) and \( X_i \) with \( (\hat{Q}_E^*) \), \( h_i \) and \( h' \) at the \( i \)th stage of the \( N \)-stage process, the conditional expected value of the error squared over \( N \) stages is

\[
E\left( \sum_{i=1}^{N} [(Q^*_E - Q_i)^2 \mid (\hat{Q}_E^*)_i, h_i, h'] \right) = E[(Q^*_E - Q_1)^2 \mid (\hat{Q}_E^*)_1, h_1, h']
\]

\[
+ E[(Q^*_E - Q_2)^2 \mid (\hat{Q}_E^*)_2, h_2, h']
\]

\[
+ \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots
\]

\[
+ E[(Q^*_E - Q_N)^2 \mid (\hat{Q}_E^*)_N, h_N, h']
\]

(5)

In addition to the constraints, Eqs. (5)-(7) of Chapter 7, Section 2, the stochastic model requires as known

\[
(\hat{Q}_E^*)_i = K_i, \quad i = 1, 2, \ldots, N
\]

(6)

\[
p_i[(Q^*_E - Q_i)^2 \mid (\hat{Q}_E^*)_i, h_i, h']. \quad i = 1, 2, \ldots, N
\]

(7)

We now define

\( F_N(h) = \) the minimum expected value of the squared error between the desired effluent and the delivered effluent summed over the \( N \) remaining stages, and subject to Eqs. (5)-(7) of Chapter 7, Section 2 and Eqs. (1), (6), and (7) of this section, beginning in state \( h \) and using an optimal policy

and we write

\[
F_N(h) = \min_{(Q^*_E)} E \left[ \sum_{i=1}^{N} (Q^*_E - Q_i)^2 \mid (\hat{Q}_E^*)_i, h_i, h' \right]
\]

(8)
11. CONDITIONAL PROBABILITY

Using the Principle of Optimality, we obtain the recurrence relation

\[ F_N(h) = \min_{(\hat{Q}_{E})} E[(Q_E - Q_{E})^2 | (\hat{Q}_{E})_N, h_N, h'] + F_{N-1}(h') \]  

(9)

\[ F_1(h) = \min_{(\hat{Q}_{E})} E[(Q_E - Q_{E})^2 | (\hat{Q}_{E})_1, h_1, h'] \]  

(10)

where \( h' \) is the new value of \( h \) at the beginning of stage \((N - 1).\)

We observe for the stochastic model that the functional equations, Eqs. (9) and (10), have the same form as the deterministic equations, Eqs. (11) and (12), of Chapter 7, Section 2.

The computational technique for the stochastic process is very similar to that given for the deterministic model. The principal difference is the utilization of the conditional frequency distribution function to determine the minimum expected error. At stage \( k \), the level \( h \) and the projected demand \( (\hat{Q}_{E})_k \) are known. By choosing various trial values of \( Q \), and hence trial values for \( h' \) the corresponding expected values are found. One way to picture this is to consider the conditional distribution function in Fig. 1, where \( (\hat{Q}_{E}) \) and \( h \) are held constant and \( h' \) assumes the values \( a_1, a_2, \) and \( a_3 \). The selection of the proper value for \( h' \) executes the optimization via the proper distribution function.

\[ (\hat{Q}_{E})_i = \text{constant} \]

\[ h_i = \text{constant} \]

Since the conditional frequency density functions are known at each stage and do not vary with the number of operating cycles, and since the \( (\hat{Q}_{E})_i \) are known, the stochastic model computations in essence reduce to the deterministic case.
Adaptive Case

As discussed in Section 11 the adaptive process is a particular type of stochastic process in which advantage is taken of information developed in the course of the $N$-stage process to develop better predictors. The adaptive model attempts to re-evaluate itself in the light of the uncertainties in the process as they unfold and change, while the ordinary stochastic process accepts its destiny supinely, so to speak.

In particular, in the adaptive case, the probability function for each stage of the $N$-stage process is re-evaluated as the process passes through each complete cycle of $N$ stages. If the probability density functions change with the number of cycles, the adaptive model uses the most recent information. In a similar manner, the estimate of $Q_\xi$ is revised in the light of recent measurements.

In the adaptive models since the probability functions change from cycle to cycle, the state of the system must take into account not only the level in the tank, but also parameters that describe the transition from one probability distribution to the next.

11. Adaptive Models

An important class of stochastic models is the adaptive or self-learning model. These models are characterized by an "updating" procedure by which the model observes or samples the behavior of the process it simulates in order to obtain a more real representation of the process itself. Self-adaptive models are designed for processes in which the stochastic nature of the process changes sufficiently to warrant careful surveillance and reassessment.

There are a wide variety of adaptive models. We will discuss only two examples to point out some of the characterizing features. In adaptive models, since we are attempting to track changes in the stochastic behavior of the process, the history of the process is important to us. Since we are interested in controlling the process, a predictor for the future behavior of the process is also important. To handle the adaptive models, we need a way to convert past knowledge of the process behavior into a tool for predicting future behavior.

If we consider a process in which the probability distribution function changes from stage to stage, then the judicious pooling and weighting of the distributions parameters, such as the mean and variance, should serve to define the probability distribution function for the succeeding
stage. It is this pooling and weighting of the distribution parameters over the past that supplies the essential tool for prediction required for adaptive processes.

To spell out the characterizing aspects of adaptive processes, we will employ the same type of general formulation used for the deterministic and stochastic processes in Sections 2 and 3. We claim that, using the terminology of Section 3, starting with an initial state \( s_N \), and control vector \( v_N \), that the process is transformed into stochastic state vector \( z_{N-1} \):

\[
z_{N-1} = T_N(s_N, v_N)
\]

In addition, we claim that from past history, we have an \textit{a priori} distribution \( dG(s, v, z) \). The arguments in \( dG(s, v, z) \) are not subscripted in order to show that this function is a weighted compilation of past history. The \textit{a priori} distribution is altered to a new distribution \( dH \):

\[
dG(s, v, z) \rightarrow dH(s, v, z; s_N, G, v_N, z_{N-1})
\]

The \textit{a posteriori} distribution depends on the past history as embodied in the \( G \) function and its arguments, the current state, current manipulated vector, and the transformed state. The \( dH \) function in turn will serve as the \textit{a priori} distribution for the succeeding stage.

Let us now optimize the return

\[
P = \sum_{i=1}^{N} g_i(s_i, v_i)
\]

We define

\[
f_N(s_N, G(s, v, z)) = \text{the maximum expected return over } z_{N-1} \text{ over the } N \text{ remaining stages, beginning in state } s_N, \text{ with the } a \text{ priori } distribution function } \ G(s, v, z), \text{ and using an optimal policy}
\]

The Principle of Optimality yields

\[
f_N(s_N; G(s, v, z)) = \underset{v_N}{\text{Max}} \int_{-\infty}^{\infty} [g_N(s_N, v_N) + f_{N-1}(z_{N-1}, H(s, v, z; s_N, G, v_N, z_{N-1}))] [dG(s_N, v_N, z_{N-1})]
\]

\[
f_1(s_1, G(s, v, z)) = \underset{v_1}{\text{Max}} \int_{-\infty}^{\infty} \{(g_1(s_1, v_1)) dG(s_1, v_1, z_0)\}
\]

In contrast to the stochastic process, the adaptive process requires in the description of the state of the system, both the state variables \( s_N \) and the \textit{a priori} distribution function (or parameters related to it).
Equations (5) and (6) are in an awkward form for computational purposes. If we assume that the $G$ and $H$ functions belong to the same probability distribution function family, we may express this as

$$G(s, v, z) = K(s, v, z, \alpha)$$

(7)

$$H(s, v, z; s_N, G, v_N, z_{N-1}) = K(s, v, z, \beta)$$

(8)

where $\alpha$ and $\beta$ are parameters.

Equation (5) becomes

$$f_N(s_N, \alpha) = \operatorname{Max}_{v_N} \int_{-\infty}^{\infty} \left[ g_N(s_N, v_N) + f_{N-1}(s_{N-1}, \beta) \right] dK(s_N, v_N, z_{N-1}, \alpha)$$

(9)

which form can be used for computation.

The reduction from a functional in Eq. (5) to a function in Eq. (9) is a transformation commonly used in statistics. For certain distribution functions, a knowledge of the parameters suffices to determine the entire distribution. For example, in a Poisson distribution, the knowledge of the mean $\mu$ is sufficient to reconstruct the entire function. For the normal distribution, the mean $\mu$ and the variance $\sigma$ characterize the distribution. In the Bernoulli distribution, the knowledge of $p$, the probability of a successful outcome, suffices. In the characterization of the distribution by the pertinent parameters, none of the information inherent in the distribution is lost. We gain a considerable economy in thought and mathematical agility by the reduction of functionals such as $f_N(s_N, G(s, v, z))$ to functions such as $f_N(s_N, \alpha)$.

12. Adaptive Averaging Control Model

To apply these concepts of adaptive process, we will discuss the averaging control problem of Section 7. We consider here again a Bernoulli distribution where with probability $p_k$, $r_k = +R$ and with probability $q_k = 1 - p_k$, $r_k = -R$. However, in contrast to the stochastic process, we do not know for sure the actual value of $p_k$. We do, however, have an a priori distribution function for $p$, namely, $dG(p)$. We assume that this has been developed from a study of the process. In addition to the a priori distribution function, we assume that we have a way of modifying the initial a priori distribution as the process unfolds to predict future behavior. The modification of the initial a priori distribution is carried out by observing and recording the actual performance of the system.
If at a given stage it is observed that the random variable \( r_k = +R \), occurs a total of \( m \) times and that the random variable \( r_k = -R \) occurs a total of \( n \) times, this information can be used to generate a new distribution function. Specifically, the modified distribution is taken as

\[
dG_{m,n} = \frac{p^m(1-p)^n dG(p)}{\int_0^1 p^m(1-p)^n dG(p)}
\]

After one more stage, \( dG_{m,n} \) is altered to

\[
dG_{m,n} \rightarrow dG_{m+1,n} \quad \text{or} \quad dG_{m,n} \rightarrow dG_{m,n+1}
\]

In these cases, \( dG_{m,n} \) correspond to \( dG \) of Eq. (2), Section 11, and \( dG_{m+1,n} \) or \( dG_{m,n+1} \) correspond to \( dH \) of Eq. (2), Section 11.

Having fixed the initial \textit{a priori} distribution \( dG(p) \), all succeeding distributions are characterized by the numerical values of \( m \) and \( n \). In essence, the distribution function \( dG_{m,n} \) is characterized by \( m \) and \( n \) just as the normal distribution function is characterized by \( \mu \) and \( \sigma \).

The return function for the adaptive process (referring to Section 7) reads as

\[
F_1(c_1, c_2, m, n) = \min_{w_1} [(c_1^2 + c_2^2 + aw_1^2) \Delta
\]

\[
+ p_{m,n}F_{N-1}(c_1 + c_2\Delta, c_2 + \{w_N + R - \mu(c_1^2 - 1)c_2 - c_1\} \Delta)
\]

\[
+ (1-p_{m,n})F_{N-1}(c_1 + c_2\Delta, c_2 + \{w_N - R - \mu(c_1^2 - 1)c_2 - c_1\} \Delta)]
\]

\[
F_1(c_1, c_2, m, n) = \min_{w_1} [(c_1^2 + c_2^2 + aw_1^2) \Delta]
\]

The quantity \( p_{m,n} \) is the expected value of \( p \) taken with respect to \( dG_{m,n} \):

\[
p_{m,n} = \int_0^1 p \ dG_{m,n}(p) = \frac{\int_0^1 p^{m+1}(1-p)^n dG(p)}{\int_0^1 p^m(1-p)^n dG(p)}
\]

The expressions, Eqs. (1) and (3)-(5), for this specific case illustrate the general concepts embodied in Eqs. (7)-(9) of Section 11. Even with the reduction of the distribution equation to two parameters, the evaluation of \( F_1(c_1, c_2, m, n) \), a function of four arguments, is a task of some magnitude. Further steps must be taken to reduce the dimensionality of the problem. This can be done by using some of the devices previously discussed. Another possibility to look for is the occurrence of an asymptotic value of \( p_{m,n} \). After many stages, \( p_{m,n} \) may converge to a fixed value. If this should occur, the adaptive process reduces to the stochastic process.
13. Adaptive Catalyst Replacement Problem

In Chapter 2, Section 5, we presented a deterministic version of the catalyst replacement problem. This problem is now set forth as an adaptive control problem for two classes of problems. In Class 1, the distribution functions are known. In the Class 2 model, the distribution functions are unknown.

Stochastic Models

In the stochastic models, the element of uncertainty is associated with the $c_i$ terms of the conversion expression, Eq. (11) of Chapter 2, Section 5. Owing to differences in manufacture of the catalyst, differences in the weight of the catalyst used at each replacement, differences in the efficiencies of regeneration of the catalyst, and channeling through the catalyst bed, the behavior of the catalyst varies from cycle to cycle and within a cycle. A cycle is the period of time between regenerations. The vagaries of the catalyst performance are a frustration and a challenge (not to mention a reality) to the plant operator who seeks to optimize the process. The operator in contending with the indeterminate nature of the catalyst must lower his sights from maximizing the profit as in the deterministic model where the $c_i$ terms are known, and settle for maximizing the expected profit in the stochastic model. This is the best that he can do in the light of uncertain and indeed incomplete information about the catalyst.

In the stochastic process, as in the deterministic process, we desire to determine the optimum profit over the entire $N$-stage process, the catalyst replacement or regeneration policy, and the temperature and flow rate–time function.

Formulation of Stochastic Models

In contrast to the deterministic model, the state of the system is determined by four variables $S$, $c_1$, $c_2$, and $c_3$ where the $S$ is defined by Eq. (12) and the $c_i$ terms by Eq. (11), of Chapter 2, Section 5.

We now define

$$ P_N(S, c_1, c_2, c_3) = \text{the expected maximum return from an } N\text{-stage process, starting in state } (S, c_1, c_2, c_3) \text{ and pursuing an optimal policy. The quantity } N \text{ refers to the number of stages remaining; } N = 1, 2, 3, ... $$

$$ (S, c_1, c_2, c_3) = \text{the state of the system characterized by } S, \text{ the cumulative throughput of the feed material up to but not including stage } N, \text{ and by } c_1, c_2, c_3 \text{ of Eq. (11), Chapter 2, Section 5} $$
13. ADAPTIVE CATALYST REPLACEMENT PROBLEM

\[ g(S, c_1, c_2, c_3, T, F) = \text{the profit for stage } N \text{ initially in state } (S, c_1, c_2, c_3) \]

where \( T \) and \( F \) are the values of temperature and flow rate chosen for stage \( N \)

\[ h(S, c_1, c_2, c_3, F) = \text{the transformed state of the system, following the selection of } T \text{ and } F \text{ for stage } N, \text{ with the current values of the } c_i \text{ terms} \]

\[ R = \text{the cost of shut down, catalyst replacement, or regeneration; one stage is required for shut down} \]

The functional equations for the stochastic process are

\[ P_N(S, c_1, c_2, c_3) = \max_{T,F} \{ \max_{T,F} \{ g(S, c_1, c_2, c_3, T, F) + P_{N-1}(h(S, c_1, c_2, c_3, F)) \} \} - R + P_{N-1}(0, c_1, c_2, c_3) \]  

(6)

(7)

The first line of Eq. (6) is the expected maximum profit for the possibility of continuing to operate without regenerating the catalyst. The quantity \( g(S, c_1, c_2, c_3, T, F) \) is the profit for stage \( N \), and the quantity \( P_{N-1}(h(S, c_1, c_2, c_3, F)) \) is the expected maximum profit for the remaining \((N-1)\) stages starting with the catalyst in the state \( h(S, c_1, c_2, c_3, F) \).

The second line of Eq. (6) pertains to the possibility of shutting down the reactor and regenerating and/or replacing the catalyst.

Equation (7) describes the one-stage process.

In Eqs. (6) and (7), we observe that the general form of the equations for the stochastic process is the same as for the deterministic process. A recursion relationship exists between the process in stage \( N \) and stage \((N-1)\). Since the process is a stochastic one, we must recognize that the \( c_i \) terms in \( g(S, c_1, c_2, c_3, T, F) \), in \( h(S, c_1, c_2, c_3, F) \), and in \( P_{N-1}(0, c_1, c_2, c_3) \) of Eqs. (6) and (7) are not the same. During the passage from one stage to another, the \( c_i \) terms will vary depending on the stochastic process.

The fact that the state of the system is a function of four variables, \( S, c_1, c_2, c_3 \), complicates considerably any numerical evaluation of the Eqs. (6) and (7). In order to be able to predict ahead in the face of uncertainty in the values of the \( c_i \) terms, it is necessary to bring in some additional information about the stochastic nature of the \( c_i \) terms.

The method discussed below essentially eliminates the \( c_i \) terms as variables by using the "best" estimate of the \( c_i \) terms at any stage \( k \) to project ahead for the remaining \((k-1)\) stages. In essence, the stochastic process is reduced to the deterministic case.
Two Classes of Stochastic Models

We consider two classes of stochastic models, those in which the distribution of the $c_i$ terms (Class I) is known and those in which the distribution is unknown (Class II). In the first class of problems, the past history of the process is used to generate better estimates of the mean and the variance which in turn are used to predict the future of the process. In the second class of problems, there is no relationship between past and future performance. The stochastic elements of the process at the current stage are measured and for lack of better information are used to estimate the future course of the process. In other words, the most recent data are the best "guesstimate" of the future.

In both classes of stochastic processes, the profit table (developed in Chapter 2, Sections 8 and 9) must be regenerated at each stage of the process in contrast to the deterministic model where it is generated only once.

Class I (Distribution of Each $c_i$ is Known)

To illustrate more specifically the Class I model where the distribution is known, we consider the case where each $c_i$ term is normally distributed with its own mean and variance.

At the beginning of the $N$-stage process, we have a priori estimates of the mean and variance of each $c_i$. Within the 95% confidence level, each $c_i$ is bounded as

$$\bar{c}_i - 2\sigma_i \leq c_i \leq \bar{c}_i + 2\sigma_i$$

where

- $\bar{c}_i$ = the estimate of the mean for the $i$th term
- $\sigma_i$ = the estimate of the standard deviation for the $i$th term

At the beginning of the $N$-stage process, we do not know, however, what the particular values of $c_i$ are for stage $N$ or for any other stage. To initiate the calculation, we use the a priori estimate of the mean of each $c_i$ for the entire $N$-stage process in the functional equations (6) and (7) and generate the profitability table as described in Chapter 2, Section 8. In other words, by using the $c_i$ terms as constants based on this a priori estimate, the stochastic model of Eqs. (6) and (7) is reduced to the deterministic case of Eqs. (5) and (6) of Chapter 2, Section 7. Because this model is a self-adaptive model, we can take advantage of newly acquired information for our predictive computations. During stage $N$ of the $N$-stage model, measurements are made in the plant to evaluate the $c_i$ terms. The newly generated $c_i$ terms are pooled with the previous data on the $c_i$ terms to generate more up-to-date estimates of the mean.
and the variance. The most recent estimates of the $\bar{c}_t$ terms are then used to regenerate the profitability table for the remaining $(N - 1)$ stages. This procedure is continued for the entire $N$-stage process. The rationale for this method is simply that the average value of each $c_t$ term is the most likely value and, therefore, the most reasonable to use for prognostication.

The computational procedure for the self-adaptive process is somewhat similar to the deterministic model in that the $c_t$ terms used are constant throughout the remaining stages for each generation of the profitability table. The self-adaptive process, on the other hand, is distinguished from the deterministic model by the fact that the means of the $c_t$ terms are reassessed at each stage as more and more information about the process is accumulated. In this method, we never actually use the values of the $c_t$ terms currently existing, but always use average values of the $c_t$ terms based on all the data available.

There are a number of ramifications of this self-adaptive model with known frequency distribution. In the “Discussion” section, we will comment on some of these.

Sensitivity Evaluation

In the Class I stochastic model, we use the most up-to-date estimate of the mean to predict the future course of the process. We do not know on entering any stage what the specific values of the $c_t$ terms are or the specific values of the $c_t$ terms for the remaining stages of the process.

It is quite possible that the $\bar{c}_t$ terms may take on such values that the length of the operating cycle may vary and accordingly the catalyst replacement or regeneration policy may also vary as the profitability table is recalculated. To explore this possibility prior to actually operating the process, we can evaluate the functional equations for a variety of conditions where the $c_t$ terms take on certain prescribed values which should bracket both the range of profitability and the range of catalyst replacement or regeneration policy.

To be specific, we shall evaluate the functional equations for the $c_t$ terms arbitrarily set at all possible combinations of their highest, lowest, and average values.

If, for convenience, with the normal distribution, we set $\bar{c}_t + 2\sigma_t = +1$, $\bar{c}_t = 0$, and $\bar{c}_t - 2\sigma_t = -1$, we can write down all the possible combinations in Table 1.

The functional equations then can be evaluated for each of the 27 cases. As a consequence of this evaluation, we can determine the sensitivity of the profitability and the catalyst replacement or regeneration policy to the uncertainty in the $c_t$ terms.
Table 1

Combinations of the $c_i$ Terms

<table>
<thead>
<tr>
<th>$c_1$</th>
<th>$c_2$</th>
<th>$c_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>+1</td>
<td>+1</td>
<td>+1</td>
</tr>
<tr>
<td>+1</td>
<td>+1</td>
<td>0</td>
</tr>
<tr>
<td>+1</td>
<td>+1</td>
<td>−1</td>
</tr>
<tr>
<td>+1</td>
<td>0</td>
<td>+1</td>
</tr>
<tr>
<td>+1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>+1</td>
<td>0</td>
<td>−1</td>
</tr>
<tr>
<td>+1</td>
<td>−1</td>
<td>+1</td>
</tr>
<tr>
<td>+1</td>
<td>−1</td>
<td>0</td>
</tr>
<tr>
<td>+1</td>
<td>−1</td>
<td>−1</td>
</tr>
<tr>
<td>0</td>
<td>+1</td>
<td>+1</td>
</tr>
<tr>
<td>0</td>
<td>+1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>+1</td>
<td>−1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>+1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>−1</td>
</tr>
<tr>
<td>0</td>
<td>−1</td>
<td>+1</td>
</tr>
<tr>
<td>0</td>
<td>−1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>−1</td>
<td>−1</td>
</tr>
<tr>
<td>−1</td>
<td>+1</td>
<td>+1</td>
</tr>
<tr>
<td>−1</td>
<td>+1</td>
<td>0</td>
</tr>
<tr>
<td>−1</td>
<td>+1</td>
<td>−1</td>
</tr>
<tr>
<td>−1</td>
<td>0</td>
<td>+1</td>
</tr>
<tr>
<td>−1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>−1</td>
<td>0</td>
<td>−1</td>
</tr>
<tr>
<td>−1</td>
<td>−1</td>
<td>+1</td>
</tr>
<tr>
<td>−1</td>
<td>−1</td>
<td>0</td>
</tr>
<tr>
<td>−1</td>
<td>−1</td>
<td>−1</td>
</tr>
</tbody>
</table>

Class II (Distribution of Each $c_i$ is Unknown)

For the Class II stochastic model, the distributions of the $c_i$ terms are unknown. To begin the calculation, we assume values of the $c_i$ terms for stage $N$ of the $N$-stage process and evaluate the profitability table using the functional equations (6) and (7). Since this is a self-adaptive model, the $c_i$ terms as determined through measurements during stage $N$ are then used in the remaining $(N - 1)$ stages to regenerate the profitability table. This procedure is continued for each stage of the entire $N$-stage process. The rationale for this method is that since the distribution is unknown, the previously determined values of the $c_i$ terms bear no known relationship to one another and hence do not contribute to estimating future values of the $c_i$ terms. The use of the measured $c_i$
terms during stage \( k \) to generate the profitability table for the remaining \((k - 1)\) stages can be justified only on the basis that the most up-to-date measured values of the \( c_i \) terms appear to be the most reasonable estimate for the succeeding stages. This is a choice by default rather than by design in view of the lack of better information.

Discussion

In the Class I stochastic model, where the distribution is known, we have demonstrated the computation procedure by using the normal distribution. The computational procedure for any other known distribution can be handled in a similar manner.

An almost endless number of possible stochastic permutations can occur. For example, after a catalyst replacement or regeneration within an \( N \)-stage process, the mean and variance of the \( c_i \) terms for the new cycle may be considerably different from the previous cycles and yet the \( c_i \) terms will be still normally distributed. It is possible within a cycle or within an \( N \)-stage process that the mean and/or the variance may move with time.

Fortunately, as the information on the \( c_i \) terms is gathered from measurements made during each stage, statistical tests can be made to determine whether the mean and/or variance are changing with time and to determine whether the mean and variance after regeneration or replacement belong to the same population as the previously evaluated mean and variance. If the \( c_i \) terms move with time, then an appropriate weighting of the past data with the more recent data can be used to develop a “best” estimate of the \( c_i \) terms to be used for prognostication.

Auditing

In both classes of stochastic models discussed here, the \( c_i \) terms are evaluated by measurements of temperature, flow rate, and product composition during each stage. In Class I, the measured \( c_i \) terms are pooled with previously known and/or measured \( c_i \) terms to obtain an up-to-date estimate of the mean \( c_i \) and variance. In Class II, the \( c_i \) terms measured during stage \( k \) are used alone to predict for the remaining \((k - 1)\) stages. At the end of the \( N \)-stage process for both stochastic classes, a set of measured \( c_i \) terms has been determined for each stage. If the functional equations are then evaluated at the end of the process, using all the measured \( c_i \) terms, the profitability and the operating policy represent the very best attainable. An audit of this profitability (after the fact, so to speak) and the actual profit earned under the stochastic operation is a measure of the cost of our ignorance in the stochastic...
process. If our ignorance is expensive, an investigation of the cause of the uncertainty in the $c_i$ terms may reveal undesirable plant operating practice, such as careless weighing of the catalyst. As a consequence, some of the factors contributing to the stochastic nature of the process may be uncovered and alleviated.

**Conclusions**

We have discussed here two classes of stochastic models which can be handled as self-adaptive models for on-line computer control of the process. In the Class I model, the distributions of the $c_i$ terms are known. As the process is operated, more and more information is developed to generate even better estimates of the mean and variance of the $c_i$ terms to be used for prediction. In the Class II model, where the distribution of the $c_i$ term is unknown, the most recently measured values of the $c_i$ term is used to predict the course of the process. In the deterministic model where the $c_i$ terms are constants, the profitability table is generated only once. In contrast to the deterministic model, the stochastic models considered here require evaluation of the profitability table at each stage as more information about the process becomes available.

The computational procedure suggested essentially reduces the stochastic problem to a deterministic one. It can be carried out easily by electronic computers and makes possible the control of an operating plant with stochastic elements through the methodology of dynamic programming.

**REFERENCES**


**PROBLEMS**

1. For the stochastic allocation problem in Section 5, consider that the $g(y)$ may take one of three possibilities. With probability $p_1$, $g(y)$ assumes values of $g_1(y)$ with $y$ decreasing to $a_1y$; with probability $p_2$, it assumes the value of $g_2(y)$ with $y$ decreasing to $a_2y$; and with probability $p_3$, it assumes the value of $g_3(y)$ with $y$ decreasing to $a_3y$. In a similar manner, we state that $h(x - y)$ may assume one of three possibilities. With probability $q_1$ it assumes the value of $h_1(x - y)$ with $(x - y)$ decreasing to $b_1(x - y)$; with
probability $q_2$ it assumes the value of $h_2(x - y)$ with $(x - y)$ decreasing
to $b_2(x - y)$; with probability $q_3$, it assumes $h_3(x - y)$ with $(x - y)$
decreasing to $b_3(x - y)$.

Show that the return function is given by

$$F_N(x) = \max_{0 \leq y \leq x} \left[ p_1 q_1 (g_1(y) + h_1(x - y) + f_{N-1}(a_1 y + b_1(x - y))) + p_2 q_2 (g_2(y) + h_2(x - y) + f_{N-1}(a_2 y + b_2(x - y))) + p_3 q_3 (g_3(y) + h_3(x - y) + f_{N-1}(a_3 y + b_3(x - y))) \right]$$

$$F_1(x) = \max_{0 \leq y \leq x} \left[ \sum_{i=1}^{3} p_i q_i (g_i(y) + h_i(x - y)) \right]$$

2. For the problem described in Section 6, suppose the behavior of the "good"
catalyst is described by

$$P_t = 100 e^{-t/10}$$

and the poor catalyst by

$$Q_t = 90 e^{-t/10}$$

Letting $t = 0, 1, 2, 3, \ldots$, determine the sensitivity of the solution to
$p = 0.9, 0.8, 0.7, \ldots$ for $R = 25, 50, 100, 200$.

3. The return from a multi-stage allocation problem is given by $g(y)$ with a
distribution return function $dR(g(y), y, x)$. The cost associated with the
return is $w(y)$ characterized by a distribution function $dD(w(y), y, x)$.

The expected return for the allocation of a initial resource $x$ is given by

$$f(x) = \max_y \left[ \int_0^\infty g \ dR(g, y, x) + f(x - \int_0^\infty w \ dD(w, y, x)) \right]$$

Show that for $\int_0^\infty w \ dD(w, y, x) << x$

$$f'(x) = \max_y \frac{\int_0^\infty g \ dR}{\int_0^\infty w \ dD}$$

If $\int_0^\infty w \ dD \neq << x$, how may these results be used?

(Bellman, R., Decision making in the face of uncertainty, I. The RAND Corporation,
P-568 (September 21, 1954).)
4. A conservative speculator has chosen to invest his funds so that he maximizes the expected value of the logarithm of the final capital in an $N$-stage time process. If he starts with an initial amount $x$ and invests the quantity $y$ where $0 \leq y \leq x$, he earns the expected return $p(x + y)$ where $p$ is the probability. If he invests $y$ dollars and loses, the expected remaining capital is $q(x - y)$, where $q$ is the probability. $p + q = 1$. The logarithm of the expected value is given as

$$E(y) = p \log (x + y) + q \log (x - y)$$

Show that

$$f_N(x) = \log x + K$$

$$f_N(x) = \max_{0 \leq y \leq x} [pf_{N-1}(x + y) + qf_{N-1}(x - y)], \quad N \geq 2$$

where

$$K = \begin{cases} \log 2 + p \log p + q \log q, & p > q \\ 0, & p \leq q \end{cases}$$

and

$$f_N(x) = \text{the expected value of the logarithm of the final capital obtained from an } N\text{-stage process starting with an initial capital } x \text{ and using an optimal policy}$$

Show that the optimal policy is unique and is independent of $N$, namely

$$y = (p - q)x, \quad p > q$$

$$y = 0, \quad p \leq q$$

(Bellman, R., and Kalaba, R., On the role of dynamic programming in statistical communication theory. IRE Trans. On Information Theory IT-3, 197-203 (1957).)

5. We desire to minimize a function $J(w)$ where

$$J(w) = \int_0^T \left[ |u| + |u'| + g(w) \right] dt$$

where $g(w)$ represents the cost of control. The system is characterized by

$$u'' + \lambda(u^2 - 1)u' + u = g(w(t)) + r(t)$$

with $u(0) = c_1$ and $u'(0) = c_2$ and $r(t)$ is a random variable. We assume that $r(t) = 0$ with probability $p$ and $r(t) = 1$ with probability $(1 - p)$. Define

$$f_N(c_1, c_2) = \min E[J_N(w)]$$

where $E$ is the expectation.
Show for the discrete case that

\[ f_N(c_1, c_2) = \min_{w_0} \left[ (|c_1| + |c_2| + g(w_0)) \Delta \right. \]
\[ + pf_{N-1}(c_1 + c_2 \Delta, c_2 + \Delta(-\lambda(c_1^2 - 1)c_2 - c_1 + w_0)) \]
\[ + (1 - p)f_{N-1}(c_1 + c_2 \Delta, c_2 + \Delta(1 - \lambda(c_1^2 - 1)c_2 - c_1 + w_0)) \]
\]


6. For the adaptive version of the previous problem, we generate by Bayes' theorem a probability \( p_{m,n} \) which bases the probability estimate on the observed \( m \) zeros and \( n \) ones that the random variable \( r(t) \) has taken.

Define

\[ f_N(c_1, c_2, m, n) = \text{the minimum, over the sequential choice of the } w_k, \text{ of the expected value of } J_N(x) \text{ over the } r_k, \text{ having observed} m \text{ zeros and } n \text{ ones} \]

Show that

\[ f_N(c_1, c_2, m, n) = \min_{w_0} \left[ (|c_1| + |c_2| + g(w_0)) \Delta \right. \]
\[ + p_m f_{N-1}[c_1 + c_2 \Delta, c_2 + \Delta(-\lambda(c_1^2 - 1)c_2 - c_1 + w_0), m + 1, n] \]
\[ + (1 - p_m,n) f_{N-1}[c_1 + c_2 \Delta, c_2 + \Delta(1 - \lambda(c_1^2 - 1)c_2 - c_1 + w_0), m, n+1] \]
\[ f_0(c_1, c_2, m, n) = (|c_1| + |c_2|) \Delta \]

7. For the retarded control problem in Section 12, Chapter 6, consider the relationship between \( x_{n-1}, x_n \), the control variable \( v_n \), and a stochastic variable \( r_n \):

\[ x_{n-1} = a_0 x_n + \ldots + a_{M} x_{n+M} + v_n + r_n \]

The successive values of \( r_n \) are statistically independent. A probability density function \( p(r_n) = p_r \) is associated with \( r_n \) in the a priori description of the problem:

\[ p_r(a \leq r_n \leq b) = \int_a^b p(y) \, dy, \quad n = N - 1, N - 2, \ldots \]

The final position of a \( k \)-stage process which begins in the initial state \( S \) is given by

\[ x_0 = B_k + \sum_{j=1}^k v_j H_{-j} + \sum_{j=1}^k r_j H_{-j} \]
where the terms are described in Problem 12 in Chapter 6. Define

\[ f_k(B_k) = \min \mathbb{E} \left[ \lambda \sum_{j=1}^{k} v_j^2 + x^2_0 \right] \]

where \( \mathbb{E} \) is the expectation. Show that

(a) \[ f_k(B_k) = \min_{v_k} [\lambda v_k^2 + \mathbb{E} f_{k-1}(B_k + v_k H_{-k} + r_k H_{-k})] \]

\[ f_0(B_0) = B_0^2 \]

(b) \[ f_k(B_k) = \frac{\lambda (B_k + m P_k)}{\lambda + Q_k} + \sigma^2 \sum_{j=1}^{k} \frac{\lambda H_{-j}}{\lambda + Q_{j-1}} \]

where

\[ Q_k = \sum_{j=1}^{k} H_{-j}; \quad P_k = \sum_{j=1}^{k} H_{-j} \]

\[ m = \int_{-\infty}^{\infty} y p(y) dy; \quad \sigma^2 = \int_{-\infty}^{\infty} (y - m)^2 p(y) dy \]

(c) \[ v_k(B_k) = - \left[ \frac{B_k + m P_k}{\lambda + Q_k} \right] H_{-k} \]

What is the significance of the results when the mean, \( m \), is zero?

(Kramer, J. D. R. [19].)

8. If in the previous problem the \( r_n \) are related by a Markov process, then we have

\[ p(r_1, ..., r_n | r_{n+1}) = p(r_1, ..., r_{n-1} | r_n) p(r_n | r_{n+1}) \]

where by our counting convention \( r_{n+1} \) occurs before \( r_n \).

Define

\[ f_k(B_k, r_{k+1}) = \min \mathbb{E} \left[ \lambda \sum_{j=1}^{k} v_j^2 + x^2_0 \right] \]

where the expectation is taken with respect to \( p(r_1, ..., r_k | r_{k+1}) \).

Show that

(a) \[ f_k(B_k, r_{k+1}) = \min_{v_j} [\lambda v_j^2 + \int_{-\infty}^{\infty} \mathbb{E} \left\{ \lambda \sum_{j=1}^{k-1} v_j^2 + x^2_0 \right\} \left( p(r_k | r_{k+1}) dr_k \right) ] \]
9. STOCHASTIC PROCESSES

where the inner expectation is taken with respect to $p(r_1, ..., r_{k-1} | r_k)$

(b) $f_k(B_k, r_{k+1}) = \min_{v_k} \left[ \lambda v_k^2 + E f_{k-1}(B_k + v_k H_{-k} + r_k H_{-k}, r_k) \right]$ $f_0(B_0, r_1) = B_0$

where the expectation is taken with respect to $p(r_k | r_{k+1})$

9. Verify that the following are solutions to the previous problem:

$$f_k(B_k, r_{k+1}) = \frac{1}{\lambda + Q_k} [B_k + g_k(r_{k+1})]^2 + S_k(r_{k+1})$$

$$v_k(B_k, r_{k+1}) = -\frac{1}{\lambda + Q_k} [B_k + g_k(r_{k+1})]^2 H_{-k}$$

where

$$Q_k = \sum_{j=1}^{k} H_{-j}^2$$

$$g_k(r_{k+1}) = \int_{-\infty}^{\infty} \left[ r_k H_{-k} + g_{k-1}(r_k) \right] p(r_k | r_{k+1}) \, dr_k$$

$$g_0(r_1) = 0$$

10. In an adaptive version of the retarded control problem governed by

$$x_{n-1} = a_0 x_n + ... + a_M x_{n+M} + v_n + r_n$$

the random variable $r_n$ takes the value of $\delta$ with probability $p$ and the value of $-\delta$ with probability $(1 - p)$. Since $p$ is not known, we can use past history taking into account the values of $-\delta$ and $m$ values of $+\delta$. By Bayes' rule the estimated $p$ called $p^*$ may be found from

$$p^* = \frac{n + a}{n + a + m + b}$$

where $a/(a + b)$ is the original a priori estimate of $p$. For convenience, we use

$$\nu = a + n; \quad \mu = b + m$$

We desire to minimize the expected value of $\sum_{j=1}^{k} v_j^2 + x_0^2$ and we define

$$f_k(B_k, \mu, \nu) = \min_{v_j} E \left[ \lambda \sum_{j=1}^{k} v_j^2 + x_0 \right]$$
With the application of the control force \( v_k \) the transformations from a \( k \)-stage to a \((k - 1)\)-stage process are given by

\[
\begin{align*}
B_k \rightarrow B_k + H_{-k}v_k + H_{-k}\delta, \\
\mu \rightarrow \mu, \\
\nu \rightarrow \nu + 1
\end{align*}
\]

with probability \( p^* = \frac{\nu}{\nu + \mu} \)

\[
\begin{align*}
B_k \rightarrow B_k + H_{-k}v_k - H_{-k}\delta, \\
\mu \rightarrow \mu + 1, \\
\nu \rightarrow \nu
\end{align*}
\]

with probability \( p^* = \frac{\mu}{\nu + \mu} \)

Show that

\[
f_k(B_k, \mu, \nu) = \min_{v_k} \left[ \lambda v_k^2 + \frac{\nu}{\nu + \mu} f_{k-1}(B_k + H_{-k}v_k + H_{-k}\delta, \mu, \nu + 1) \\
+ \frac{\mu}{\nu + \mu} f_{k-1}(B_k + H_{-k}v_k - H_{-k}\delta, \mu + 1, \nu) \right]
\]

\[
f_0(B_0, \mu, \nu) = B_0^2
\]

Verify that the solution is given by

\[
f_k(B_k, \mu, \nu) = \frac{\lambda \left( B_k + \delta \frac{\nu - \mu}{\nu + \mu} P_k \right)^2}{\lambda + Q_k} + g_k(\delta, \mu, \nu)
\]

and the control forces by

\[
v_k(B_k, \mu, \nu) = -\left[ \frac{\left( B_k + \delta \frac{\nu - \mu}{\nu + \mu} P_k \right)}{\lambda + Q_k} \right] H_{-k}
\]

where

\[
Q_k = \sum_{j=1}^{k} H_{-j}; \quad P_k = \sum_{j=1}^{k} H_{-j}
\]

11. An investor with \( m \) dollars desires to maximize his return over an \( N \)-time stage period. After each investment the capital is reduced by \( b \) dollars, the investment cost. He has the option to invest or not to invest at each stage. If we define

\[
f_k(m) = \text{the maximum expected discounted return starting with } m \text{ dollars over the } N \text{ remaining stages} \\
y = \text{the return from the investment} \\
a = \text{discount factor} \\
y_n(m, b) = a f_{n-1}(m) - a f_{n-1}(m - b)
\]
the investment return possibilities are given by

(a) \( y + f_{n-1}(m - b) \); accept investment, if \( y > y_n(m, b) \)
(b) \( af_{n-1}(m) \); reject investment, if \( y < y_n(m, b) \)

Show that

\[
\begin{align*}
    f_n(m) &= \max \{ y + af_{n-1}(m - b), af_{n-1}(m) \}; \quad b \leq m \\
    f_n(m) &= af_{n-1}(m), \quad b > m
\end{align*}
\]

If \( \varphi_n(y, b) \) is the joint probability density function of \( y \) and \( b \), show that \( f_n(m) \) and the decision criterion \( y_n(m, b) \) can be found by

\[
f_n(m) = \int_0^m \int_{-\infty}^\infty \max \{ y + af_{n-1}(m - b), af_{n-1}(m) \} \varphi_n(y, b) \, dy \, db \\
+ af_{n-1}(m) \int_m^\infty \int_{-\infty}^\infty \varphi_n(y, b) \, dy \, db
\]

(Fisher, J. L. [16].)

12. Suppose in the previous problem that the investor has an alternative of investing all or part of his money in an ever-available prospect, such as government bonds, above and beyond the previously described investment opportunities. The rate of return for this is called \( y^* \). If the investor has larger funds available and not enough investment possibilities, he, rather than let these funds be discounted, may set a maximum amount that he would be willing to hold until the next investment possibility. Let \( m^*_{n-1} \) be the maximum funds to be held until next decision making time. The investor, on the other hand, may have such limited funds that if he passes up investment opportunity, his return in the future would be greatly discounted. Let \( m'_{n-1} \) be the minimum amount of funds that the investor would hold until the next decision making time. Let us observe that, for \( y_n(m, b) \) to be optimal, \( y_n(m, b) \geq by^* \). Show that this is true if \( af_{n-1}(m - b) \) satisfies

\[
\frac{d}{dm} [af_{n-1}(m)] \geq y^*
\]  \( (1) \)

\[
\frac{af_{n-1}(m)}{m} \geq y^*
\]  \( (2) \)

Show that \( m^*_{n-1} \) and \( m'_{n-1} \) are defined by

\[
\frac{d}{dm} [af_{n-1}(m)]_{m=m^*_{n-1}} = y^*
\]  \( (3) \)

\[
\frac{af_{n-1}(m^*_{n-1})}{m^*_{n-1}} \geq y^*
\]  \( (4) \)
PROBLEMS 447

For \( m < m^* \), show that \( f_n(m) \) is given by

\[
\frac{df_n}{dm} = \left[ af_{n-1}(m) \right]_{m=m_{n-1}} > y^* \quad (5)
\]

\[
\frac{af_{n-1}(m_{n-1}^*)}{m_{n-1}^*} = y^* \quad (6)
\]

Show for \( m - b \geq m_{n-1}^* \)

\[
f_n(m) = \text{Max} \left[ y + (m - b - m_{n-1}^*)y^* + af_{n-1}(m_{n-1}^*), (m - m_{n-1}^*)y^* + af_{n-1}(m_{n-1}^*) \right] \quad (7)
\]

\[
y_n(m, b) = by^* \quad (8)
\]

and for \( m - b < m_{n-1}^* \)

\[
f_n(m) = \text{Max} \left[ y + (m - b)y^*, af_{n-1}(m) \right] \quad (9)
\]

\[
y_n(m, b) = af_{n-1}(m) - (m - b)y^* \quad (10)
\]

For \( m \leq m_{n-1}^* \), show that \( f_n(m) \) is given by

\[
f_n(m) = \int_0^{m-m'} \int_{y_n(m, b)}^{\infty} \left[ y + af_{n-1}(m - b) \right] \varphi_n(y, b) \, dy \, db \\
+ \int_{m-m'}^m \int_{y_n(m, b)}^{\infty} \left[ y + (m - b)y^* \right] \varphi_n(y, b) \, dy \, db \\
+ af_{n-1}(m) \left[ \int_m^{m'} \int_{-\infty}^{\varphi_n(y, b)} \varphi_n(y, b) \, dy \, db + \int_0^m \int_{y_n(m, b)}^{\varphi_n(y, b)} \varphi_n(y, b) \, dy \, db \right] \quad (11)
\]

13. Referring to Problems 11-12 for an infinite number of time periods where \( \varphi_n(y, b) = \varphi(y, b) \) and \( m \leq m^* \), prove

\[
f(m) = \text{Max} \left[ y + af(m - b), af(m) \right] \\
y(m, b) = a[f(m) - f(m - b)], \quad m - b \geq m' \\
f(m) = \text{Max} \left[ y + (m - b)y^*, af(m) \right] \\
y(m, b) = af(m) - (m - b)y^*, \quad m - b < m' \\
f(m) = \int_0^{m-m'} \int_{y(m, b)}^{\infty} \left[ y + af(m - b) \right] \varphi(y, b) \, dy \, db \\
+ \int_{m-m'}^m \int_{y(m, b)}^{\infty} \left[ y + (m - b)y^* \right] \varphi(y, b) \, dy \, db \\
+ af(m) \left[ \int_m^{m'} \int_{-\infty}^{\varphi(y, b)} \varphi(y, b) \, dy \, db + \int_0^m \int_{y(m, b)}^{\varphi(y, b)} \varphi(y, b) \, dy \, db \right]
\]
14. Referring to Problems 11-13 suppose we may or may not receive an investment opportunity for each time stage. Let us define

\[
\begin{align*}
p &= \text{the probability of receiving an investment opportunity at the beginning of a time period} \\
q &= 1 - p = \text{the probability of not receiving an investment opportunity} \\
F(m) &= \text{maximum expected discounted return immediately after decision making, assuming there is a probability } p \text{ of receiving an offer at the beginning of any period}
\end{align*}
\]

Show

\[
E[F(m)] = p[a(m)] + pq[a^2f(m)] + \ldots + pq^{k-1}[a^kf(m)]
\]

\[
+ \ldots = \frac{p}{1 - qa} [a(m)]
\]

Justify that if, in the previous equations, \( a \) is replaced by \( pa/(1 - qa) \), the previous equations are still applicable.

15. Referring to Problem 14 suppose in addition to the initial investment funds \( m \), the investment funds increase at the rate of \( c \) per stage. Considering that investment possibilities may present themselves at each stage with a probability of \( p \), show that for \( m - b \geq 0 \)

\[
f(m) = \text{Max} \left\{ y + paf(m - b + c) + pqaf(m - b + 2c) \right. \\
+ pq^2af(m - b + 3c) + \ldots \\
+ pq^{k-1}af(m - b + kc) + p(aq)^{k}[c + af(m*)] \\
+ pq^{k+1}a^{k}[c + ac + a^2f(m*)] + \ldots , \\
\right.
\]

\[
paf(m + c) + pq^2f(m + 2c) + \ldots + pq^{r-1}af(m + rc) \\
+ p(aq)^{r}[c + af(m*)] + pq^{r+1}a^{r}[c + ac + a^2f(m*)] + \ldots \\
\]

or

\[
f(m) = \text{Max} \left\{ y + p \sum_{i=1}^{k-1} a^{i-1}f(m - b + ic) + \frac{(aq)^{k-1}}{1 - aq} [paf(m*) + cqa], \\
\right.
\]

\[
\left. p \sum_{i=1}^{r-1} a^{i-1}f(m + ic) + \frac{(aq)^{r-1}}{1 - aq} [paf(m*) + cqa] \right\}
\]

where \( k = (m* - m + b) \) and \( r = (m* - m)/c \),
and for \( m - b < 0 \)

\[
f(m) = p \sum_{i=1}^{r-1} a^{i}q^{i-1} f(m + ic) + \left[ \frac{(aq)^{r-1}}{1 - aq} \right] [paf(m^*) + cqa]
\]

16. For a second-order system characterized by

\[
\frac{dx^2}{dt} + a \frac{dx}{dt} + bx = f(t) + r(t)
\]

where \( f(t) \) is the forcing function and \( r(t) \) is the noise function, minimize the performance index \( J(x) = \int_{0}^{T} x^2 \, dt \).

Using the initial conditions

\[
x(0) = c_1 \quad \text{and} \quad x'(0) = c_2
\]

and limiting \( |f(t)| \leq A \), show that the functional equations are

\[
F_{k+1}(c_1, c_2) = \min_{j_k} \left[ c_1^2 A + F_k(c_1 + c_2 \Delta, c_2 + (f_0 + r_0 - ac_2 - bc_1) \Delta) \right]
\]

\[
F_1(c_1, c_2) = c_1^2 A
\]

(Schlager, K. J., and Higgins, T. J., Dynamic programming in design of feedback control systems. AIEE DP-60-711, AIEE Great Lakes District Meeting, Milwaukee, April 27–29, 1960.)

17. In a chemical analysis of an intermediate chemical product a series of reactions must be carried out and tested sequentially in the laboratory to determine the final product yield. To assure that each step in this procedure is executed properly, at each stage one or more duplicate tests may be carried out. To determine the number of duplicate tests at each stage for a given expenditure show that the following formulation is adequate. Let

\[
p_j(x_j) = \text{the probability of a successful performance at the } j \text{th reaction stage if } (1 + x_j) \text{ repetitive tests are carried out at the } j \text{th stage (known numbers)}
\]

\[
c_j = \text{cost of the } j \text{th stage of testing}
\]

\[
c = \text{total cost of testing, so that } c \geq \sum_{j=1}^{N} c_j x_j
\]

\[
x_j = 0, 1, 2, ...
\]
If the objective is to maximize $\prod_{j=1}^{N} p_j(x_j)$ for total cost of testing, $c$, show that

$$f_N(c) = \max_{x_N} [p_N(x_N) f_{N-1}(c - c_N x_N)], \quad N = 2, 3, ...$$

$$f_1(c) = p_1(x_1) = p_1 \left( \frac{c}{c_1} \right)$$

where

$f_N(c)$ = the maximum probability of successful performance over the $N$ stages for a total cost $c$ and using an optimal policy

(Bellman, R., Combinatorial processes and dynamic programming. The RAND Corporation, P-1284 (February 24, 1958).)
Numbers in parentheses are reference numbers and indicate that an author's work is referred to although his name is not cited in the text. Numbers in italics indicate the pages on which the complete references are listed.

Amundson, N. R., 64, 66, 68, 249, 333, 335, 336
Aoki, M., 280, 295, 438
Aris, R., 64, 66, 68, 69, 118, 121(1,2), 144, 237, 249, 333, 335, 336, 337, 339, 340
Arrow, K. J., 360(2), 385, 386, 393, 400, 401, 403, 404, 406, 407, 408, 409, 410
Ash, M., 280
Barnes, J. L., 282
Beckwith, R. E., 144, 237
Bellman, R., 8(1), 10, 31(2), 32, 64, 69, 71, 72, 73, 78, 89(5, 7, 20), 110, 145, 146, 149, 150, 151, 153, 155, 195(16), 237, 238, 243, 245, 247, 272(6), 280, 281, 296, 386, 399, 438, 439, 440, 441, 442, 450
Bernholtz, B., 386
Bertram, J. E., 333
Bickley, W. G., 147
Bishop, A. B., 388
Bliss, G. A., 81(30), 146
Blum, E. D., 338
Bolza, O., 81(28, 29), 146
Bond, A. F., 390
Boydston, R. E., 146
Brown, G. S., 281
Campbell, D. P., 281, 341
Cartaino, D. F., 146, 242
Chambers, J. C., 390
Chang, S. S. L., 281
Courant, R., 81(33), 146
Dantzig, G. B., 238
Davies, O. L., 21(4), 32
Dranoff, J. S., 64
Dreyfus, S. E., 32, 33, 110, 145, 146, 147, 151, 237, 238, 241, 242, 248, 281, 349(15), 386
Dvoretzky, A., 386
Eckman, D. P., 141(35), 146
Fabian, T., 386
Fadden, E. J., 141(36), 146
Fang, Y., 275, 281
Fisher, J. L., 386, 439, 446
Ford, L. R., 124(37), 146
Freimer, M., 281, 439
Gardner, M. F., 282
Gass, S. I., 238
Glucksberg, L., 89(20), 145, 237, 281, 386, 399
Gluss, B., 391, 394
Gomory, R., 238
Goodman, L., 141(38), 146
Gould, L. A., 146, 282
Graham, L. J., 386
Gross, O., 89(20), 145, 237, 281, 386, 399
Harris, T. E., 385
Higgins, T. J., 275, 281, 282, 283, 449
Hilbert, D., 81(33), 146
Hildebrand, F. B., 81(40), 146, 215(41), 238
Ince, E. L., 124(41), 146
Jefferson, J. T., 57, 64
Juncosa, M. L., 32
Kaiser, J. F., 282
Kalaba, R. E., 32, 64, 69, 238, 280, 281, 285, 439, 441
Kallay, N., 64, 76, 77
Kalman, R. E., 249, 333, 334, 335, 336
Karlin, S., 360(2), 386, 393, 400, 401, 403, 404, 406, 407, 408, 409, 410
Katz, S., 141(43), 146
Kesner, O., 386
Kiefer, J., 386
Kinney, R., 141(43), 146
Kipiniak, W., 146
Köepcke, R. W., 333, 334
Kramer, J. D. R., 282, 286, 288, 439, 443
<table>
<thead>
<tr>
<th>Author</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lane, J. W.</td>
<td>439</td>
</tr>
<tr>
<td>Lapidus, L.</td>
<td>249, 313, 333, 335, 336</td>
</tr>
<tr>
<td>Laspe, C. G.</td>
<td>21(8), 32, 333</td>
</tr>
<tr>
<td>Lee, E. B.</td>
<td>282</td>
</tr>
<tr>
<td>Lefkowitz, I.</td>
<td>141(35), 146</td>
</tr>
<tr>
<td>Levy, J.</td>
<td>402</td>
</tr>
<tr>
<td>Little, J. D. C.</td>
<td>386</td>
</tr>
<tr>
<td>Mahoney, J. D.</td>
<td>146, 238</td>
</tr>
<tr>
<td>Maitra, K. K.</td>
<td>282</td>
</tr>
<tr>
<td>Marschak, J.</td>
<td>385</td>
</tr>
<tr>
<td>Miele, A.</td>
<td>89(44-46), 146</td>
</tr>
<tr>
<td>Mills, H. D.</td>
<td>398</td>
</tr>
<tr>
<td>Mitten, L. G.</td>
<td>64</td>
</tr>
<tr>
<td>Morin, F.</td>
<td>386</td>
</tr>
<tr>
<td>Newton, G. C.</td>
<td>282</td>
</tr>
<tr>
<td>Parzen, E.</td>
<td>439</td>
</tr>
<tr>
<td>Richardson, J. M.</td>
<td>145</td>
</tr>
<tr>
<td>Roberts, S. M.</td>
<td>21(8), 32, 33, 65, 146, 238, 333, 439</td>
</tr>
<tr>
<td>Rockwell, T. H.</td>
<td>388</td>
</tr>
<tr>
<td>Rudd, D. F.</td>
<td>64, 66, 68, 338</td>
</tr>
<tr>
<td>Sandelien, J. F.</td>
<td>146, 282</td>
</tr>
<tr>
<td>Sarachik, P. E.</td>
<td>333</td>
</tr>
<tr>
<td>Sasieni, M. W.</td>
<td>386</td>
</tr>
<tr>
<td>Scarf, H.</td>
<td>360(2), 386, 393, 400, 401, 403, 404, 406, 407, 408, 409, 410</td>
</tr>
<tr>
<td>Schlager, K. J.</td>
<td>282, 283, 449</td>
</tr>
<tr>
<td>Shapiro, E.</td>
<td>313(6), 333</td>
</tr>
<tr>
<td>Shapiro, S.</td>
<td>313(6), 333</td>
</tr>
<tr>
<td>Shelson, W.</td>
<td>386</td>
</tr>
<tr>
<td>Simpson, K. F., Jr.</td>
<td>395</td>
</tr>
<tr>
<td>Sneddon, I. N.</td>
<td>147</td>
</tr>
<tr>
<td>Sokolnikoff, E. S.</td>
<td>238</td>
</tr>
<tr>
<td>Sokolnikoff, I. S.</td>
<td>238</td>
</tr>
<tr>
<td>Stevens, W. F.</td>
<td>64</td>
</tr>
<tr>
<td>Stillman, R. E.</td>
<td>313(6), 333</td>
</tr>
<tr>
<td>Szego, G.</td>
<td>238</td>
</tr>
<tr>
<td>Temple, G.</td>
<td>147</td>
</tr>
<tr>
<td>Tou, J. T.</td>
<td>282, 289, 290</td>
</tr>
<tr>
<td>Valentine, F. A.</td>
<td>88, 147</td>
</tr>
<tr>
<td>Wanninger, L. A.</td>
<td>64</td>
</tr>
<tr>
<td>Weinstock, R.</td>
<td>81(52), 147</td>
</tr>
<tr>
<td>Westbrook, G. T.</td>
<td>65</td>
</tr>
<tr>
<td>White, D. J.</td>
<td>397</td>
</tr>
<tr>
<td>Wolfowitz, J.</td>
<td>386</td>
</tr>
<tr>
<td>Yardenii, A.</td>
<td>386</td>
</tr>
<tr>
<td>Zachary, R. A.</td>
<td>371, 386</td>
</tr>
</tbody>
</table>
Subject Index

Adaptive processes, auditing, 437, 438
averaging control, 430, 431, 442, 449
catalyst replacement, 432-438
Class I, 434-436
Class II, 436, 437
general, 428-430
tank control, 428
time lag terminal control, 444, 445
Allocation problems, 39-64
classification of, 39
dynamic programming and differential calculus, 177-179
see approximation in function space
see approximation in policy space
see cross-current extraction
see efficiency of units in tandem
see liquid pipelines
see multi-reactor systems
see multi-unit chemical plant
see quasi-optimal policy approximation
see stochastic processes
Analytical return functions, 194-197
Approximation in function space, 6, 7, 181, 182, 186-189, 240
Approximation in policy space, 7, 181-185, 226, 227, 240, 338
alternative policy, 194, 195
Averaging control, 253-260, 283, 284
general formulation, 253-255
see adaptive processes
see tubular reactor
Bang-bang control, 295
Batch reaction,
terminal control, 338
time minimization, 128-142
numerical example, 134-139
Brachistochrone, 9, 124-128
by calculus of variations, 125, 126
comparison of solution methods, 142, 143
by dynamic programming, 127, 128
Calculus, 2, 11
Calculus of variations, 1, 2, 8, 9, 11, 79-82, 285
difficulties in, 83
equality constraints, 84, 85
example of inequality constrained problem, 93-97
inequality constraints, 85-89
isoperimetric constraints, 84
Lagrangian multiplier, 84
linearity, 92, 93
Max \int_0^\infty F(x, y) \, dt, 106, 107
nonanalytic functions, 93
numerical evaluation of
\[ f_T = \text{Max} \left[ F(c, v) + G(c, v) f_c \right], \]
110-114
and ordinary maxima and minima, 83
two-point boundary value problems, 90-92
Calculus of variations and dynamic programming, 19-144
see variational problems
Catalyst replacement/regeneration, 8, 12, 19-32, 38
computational aspects for Model II, 25-32
infinitely long process, 33
Model I—simplest case, 22, 23
Model II—N-stage process, 23-32
policy, 26
see adaptive processes
see calculus of variations, example of inequality constrained problem
see stochastic processes
Chemical engineering,
absorption, 37
filter, 37
ion exchanger, 37
heat exchanger, 37, 38, 69, 70
thermal tubular reactor, 37
see catalyst replacement/regeneration
see compressor
see control, absorber with quadratic performance index
see control, of chemical reactor
see cross-current extraction
see multi-reactors
see multi-unit chemical plant
see nuclear power system
Combinatorial method, 160, 161, 204

453
Combinatorial problems, 9, 10, 160
Compressor, 69
Computational aspects of dynamic programming, 157-236
  general computation process, 163-166
Computer control,
  of batch process, 141
  hydrogenation process, 141
Concave function, 196, 197
Constraints, 22
  devices for handling inequality, 87-89
    augmented objective function, 88, 89
    bliss, 87
    positivity, 87, 88
    special function, 89
  Neyman-Pearson lemma, 89
  inequality, 9, 85-89, 117
Continuous functional equations, 6
Continuous process, 6
Control,
  absorber, with quadratic performance index, 323-326
  alternate method of Kalman, 330-333
  of chemical reactor, 117-121
  numerical example, absorber with quadratic performance index, 326-330
  with quadratic performance index, 290-294, 317-323, 333, 334
  see adaptive processes
  see averaging control
  see stochastic processes
  see terminal control
  see time lag control
Convex function, 196, 197
Cross-current extraction, 47-50, 66-68
  alternate formulation, 49, 50
Deterministic processes,
  general formulation, 412, 413
Differential calculus, 83, 248
Dimensionality,
  devices for handling high dimensionality, 169, 174-177
  high, 10
  multi-dimensional problems, 168-169
  multi-dimensional, isoperimetric problem, 205, 206
  reduction of, 10
  reduction of, by linearization, terminal control, 220-226
two-dimensional problems, 166-168
  see mechanical quadrature
  see reduction of, linear programming
Discrete process, 5
Discrete set, 230-233
  extension to linear programming, 231-233
  maximization over, 230, 231
Dynamic programming,
  advantages of, 8, 9, 140-144, 230
  and calculus of variations, 79-144,
    summing up, 143, 144
  computational aspects of, 12, 157-236
    reasons for, 158-160
  and constraints, 144
  definition of, 2
  disadvantages of, 10
  and linearity, 144
  method of attack, 7, 8
  and nonanalytical forms, 144
  nonlinear partial differential equation, 98, 101-103, 105, 143, 144, 221, 225
  and two-point boundary problems, 143
  versus combinatorial approach, 160
  see combinatorial method
Efficiency of units in tandem, 62-64
Equipment replacement, 12, 13-19, 32-36
  classes of, 11
  with improving technology, 33-35
  infinitely long process, 17-19, 33
  with used equipment, 35, 36
Erdman corner conditions, 87
Euler-Lagrange equations, 9, 80-87, 90, 92, 93, 95-97, 109, 147-8
  derivation by dynamic programming, 108-110.
Expanding grid, 166, 235, 310
Expectation, 415
Feedback control,
  and multi-stage decision process, 253
Functional equation, 3
  additive form, 62, 64
  product form, 63, 64
  numerical example, 74-76
Grid, 25, 26, 111, 113, 114, 131-134, 136, 139, 140, 159, 164, 165, 166-168
  see expanding grid
  see step-by-step grid generation
Heat exchangers, 69, 70
Homogeneity, 68, 233-236
Hydroelectric scheduling, 407

Imbedding, 5, 104, 108, 116
Infinitely long process, 17-19, 33
Infinite-stage process, 165, 189-191
Initial value problems, 9
see dynamic programming, nonlinear partial differential equation
Integral equation, 155
Interpolation, 26, 164
Inventory,
Arrow-Harris-Marschak model, 407-409
continuous model, 355-357
evaluation of, 357-360
numerical example of, 360, 361
with setup charges, 361, 362
control and production control, 369-373
discrete model, 362, 363
effect of errors in costs, 390-393
with horizons, 397, 398
with increasing demand, 402
maximum profit, production and inventory control, 373-376
maximum rate of return, 376-378, 390
minimum cost, 390-394, 399
in plant, 378-380, 387
and production scheduling, 409, 410
purchasing on fluctuating market, 366-369
simple ordering policy, 400, 401
with stochastic input, 363-365
time lag, deferred payment of demand model, 403-406
time lag, lost sales model, 403, 404
two-commodity, 398, 399
Investment, 73, 77, 78, 441
Isoperimetric constraint, 84, 85, 148, 220, 222, 226, 259, 260, 262, 263, 266, 271, 307, 311

kth best policy, 202-205, 206-213, 239
numerical example, 209-213

in linear programming, 235, 236
numerical, counter example, 202-205
numerical example, 199-202
Least sum, 240-242
Least time, 128-142, 160-163, 283
stochastic, 294, 295
Legendre condition, 147
Legendre-Gauss, 214-216
Legendre polynomial, 215-219
Linear approximations, successive, 226, 227
Linearization,
around equilibrium, 313-315
Linear programming, 10, 248
and dynamic programming, 229-235
reduction of dimensionality, 233-235
see discrete set, extension to linear programming
Liquid cross-current extraction, 40, 47-50
Liquid-liquid extraction, 335, 336
Liquid pipeline,
minimization of pumping costs, 55-62
generalization of, 59-62
Mechanical quadrature, 214-219
and dynamic programming, 216-219
Method of characteristics, 119-121
Multi-reactor system, 39, 65, 66, 239
feed distribution over reactors and time, 45-47
parallel reactors, 40, 41
parallel reactors with feed forward, 41-43
series reactor system, 43, 44
Multi-stage decision process, 2, 3, 9, 20
two examples of, 2, 3
with recycle, 337, 338
Multi-unit chemical plant, 50-54
Minimum of maximum deviation, 264, 265, 295, 296
Newton-Cotes, 214, 215
Nonlinear partial differential equation, see dynamic programming
Nuclear power system, 74-77
Numerical examples, combinatorial versus dynamic programming, 160-163
equipment replacement, 15-17
see batch reaction control
see catalyst replacement/regeneration
see control, absorber with quadratic performance index
see functional equations product form
see inventory, continuous model
see kth best policy
see optimal design of control systems
see step-by-step grid generation
see successive approximations, Picard's method
see warehousing problem

Optimal design of control systems, 275-280
numerical example of, 278-280

Performance index
quadratic, 316, 317, 333, 334
Planning model, 396, 397
Plant expansion, 393
Policy, 4, 8, 18
catalyst replacement, 20
Policy space,
see approximation in policy space
Present worth factor, 17-19, 356, 362, 370
Principle of optimality, 14, 24
definition of, 5

Quadratic approximations, 194, 195
Quasi-optimal policy approximation, 191-194

Reactor,
single, 44, 45
stirred tank, 334, 335, 337, 339
see batch reaction
see tubular reactor
Recursion, 14
Replacement/regeneration problems, 11-32
see catalyst replacement/regeneration
see equipment replacement
see infinitely long process
Retarded control
see time lag control

Scheduling, 388-390
Search techniques, 10
Separation factors, 50-54
Sequential testing, 449, 450
Simplex algorithm, 10, 248
Smoothing problems, 245, 246, 387
quadratic, 381-385
as variational problem, 121-124
Stages, 3
how many, 165
State of the system, 3, 40, 45, 412-416
determination of, 165
specified by distribution function, 227, 228
Step-by-step grid generation, 169-177
equation of, 170-177
numerical example of, 170-174
see grid

Stochastic matrix, 424
Stochastic processes, 411-438
allocation problem, 418, 419, 439, 440
averaging control, 421-423, 441, 442
catalyst replacement/regeneration, 419-421, 440
density functions, 423
general formulation, 414-416
investment, 445-449
tank control, 425-428
time lag, terminal control, 442-444
versus deterministic, 416, 417

Successive approximations, 6
dynamic programming, 181, 182
Picard's method, 179-181
Successive linear approximations, 226, 227

Tank control, 298-304
discussion of, 303, 304
multi-tank, 303
single tank, 298-300, 340, 341
two-tank, 301, 302
see stochastic processes

Taylor's series, 100, 104, 108

Terminal control, 260-264, 282, 285, 286
generalization of, 263, 264
linearized equations, 222-226
multi-variables, 263
in two variables, 262
see dimensionality, reduction of, by linearization
see tubular reactor

Time lag control, 266-270, 286-289
alternate approach, 270-275

Transition matrix, 289, 316

Tubular reactor
averaging control, 311-313
minimum of maximum deviation, 337
terminal control of, 304-310
Two-point boundary value problems, 9
see calculus of variations
see dynamic programming

Variational problems,
by dynamic programming,
finite analog, to Max $\int_0^T F(x, y) \, dt$, 114, 115
finite analog, to Min $\int_0^T F(y, y', t) \, dt$
116, 117
dynamic programming, point of view, 97
dynamic programming, solutions of,

Max $\int_0^T F(x, y) \, dt$, 98-102
Max $\int_0^\infty F(x, y) \, dt$, 102, 103
Max $\int_0^T F(x, y, t) \, dt$, 104-106
see smoothing problems

Warehousing problem, 247, 248, 344-355, 386, 387, 395, 396
analytical solution, 347-353
comparisons of solutions, 355
by dynamic programming, 344-346
numerical example, 354, 355
Weierstrass condition, 147