CHAPMAN & HALL/CRC
Monographs and Surveys in
Pure and Applied Mathematics 110

ITERATIVE DYNAMIC PROGRAMMING

REIN LUUS

CHAPMAN & HALLKRC
Monographs and Surveys in Pure and Applied Mathematics

Main Editors

Editorial Board

CHAPMAN & HALL/CRC
Monographs and Surveys in
Pure and Applied Mathematics 110

ITERATIVE DYNAMIC PROGRAMMING

REIN LUUS

CHAPMAN & HALL/CRC
Boca Raton London New York Washington, D.C.

Library of Congress Cataloging-in-Publication Data

Luus, Rein
Iterative dynamic programming / Rein Luus.
p. cm. -- (Monographs and surveys in pure and applied mathematics; 110)
Includes bibliographical references and index.
ISBN 1-58488-148-8 (alk. paper)
1. Dynamic programming. 2. Iterative methods (Mathematics) 1. Title. 11. Chapman & HalVCRC monographs and surveys in pure and applied mathematics; 110.

QA402.5 .L88 ZOO0
519.7'03-dc21 99-058886
CIP

Direct all inquiries to CRC Press LLC, 2000 N.W. Corporate Blvd., Boca Raton, Florida 33431.

International Standard Book Number 1-58488-148-8
Library of Congress Card Number 99-058886
Printed in the United States of America 1 2 3 4 5 6 7 8 9 0
Printed on acid-free paper

This book is dedicated to
Professor Rutherford Aris

Contents

About the author

Rein Luus received his B.A.Sc. degree in Engineering Physics in 1961 and M.A.Sc. in Chemical Engineering in 1962 from the University of Toronto, and an A.M. degree in 1963 and Ph.D. degree in 1964 from Princeton University. In 1964, he was granted a Sloan Postdoctoral Fellowship, and during his postdoctorate studies at Princeton University, he wrote, with Professor Leon Lapidus, the book *Optimal Control of Engineering Processes*. In 1965, he joined the University of Toronto where he is currently Professor of Chemical Engineering.

Professor Luus has published more than 100 papers in scientific journals. A large number of these papers deal with his recent developments in iterative dynamic programming. He has served as a consultant for Shell Canada, Imperial Oil, Canadian General Electric, Fiberglas Ltd., and Milltronics. He spent a sabbatical year in the research department at Steel Company of Canada, doing mathematical modelling, simulation, and data analysis. In 1976, he was awarded the Steacie Prize, and in 1980 the ERCO award. He has devoted more than 38 years to his profession as a researcher and teacher.

Preface

Dynamic programming, developed by Richard Bellmann, is a powerful method for solving optimization problems. It has the attractive feature of breaking up a complex optimization problem into a number of simpler problems. The solution of the simpler problems then leads to the solution of the original problem. Such stage-by-stage calculations are ideally suited for digital computers, and the global optimum is always obtained. The drawbacks consisting of the *curse of dimensionality* and *menace of the expanding grid*, coupled with interpolation problems, have limited dynamic programming to solving optimal control problems of very low dimension.

To overcome these limitations of dynamic programming, I suggested ten years ago to use dynamic programming in an iterative fashion, where the interpolation problem is eliminated by using the control policy that was optimal for the grid point closest to the state, and by clustering the grid points closer together around the best value in an iterative fashion. Such a scheme, however, was computationally not feasible, since a two-dimensional optimal control problem with a scalar control took over an hour to solve on the Cray supercomputer. However, a slight change made the computational procedure feasible. Instead of picking the grid points over a rectangular array, I generated the grid points by integrating the state equations with different values of control. For that two-dimensional optimal control problem the computational effort was reduced by a factor of 100, and the dimensionality of the state vector no longer mattered. This led to what now is termed *iterative dynamic programming.* In iterative fashion, dynamic programming can now be used with very high-dimensional optimal control problems. The goal of this book is to give a working knowledge of iterative dynamic programming (IDP), by providing worked out solutions for a wide range of problems.

A strong background in mathematical techniques and chemical engineering is not essential for understanding this book, which is aimed at the level of seniors or first-year graduate students. Although many of the examples are from chemical engineering, these examples are presented with sufficient background material to make them generally understandable, so that the optimal control problems will be meaningful.

In Chapter 1, the basic concepts involving mathematical models and solution of sets of nonlinear algebraic equations are presented. In Chapter 2, two steady-state optimization procedures that I have found very useful and which provide the necessary links to ideas pertaining to iterative dynamic programming are presented and illustrated. In Chapter 3, application of dynamic programming is illustrated with several examples to give the reader some appreciation of its attractive features. In Chapter 4, I present the basic ideas underlying iterative dynamic programming.

In Chapter 5, different ways of generating allowable values for control are examined. In Chapter 6, I examine in a preliminary fashion the effects of the parameters involved in IDP. Such evaluation of the parameters is continued throughout the book. In Chapter 7, it is shown that the use of piecewise linear continuous control leads to great advantages when the control policy is smooth. Comparison of IDP with solution of the Riccati equation for a quadratic performance index shows the advantages of IDP. In Chapter 8, it will become obvious to the reader that the optimal control of time-delay systems presents no real difficulties. In Chapter 9, the use of variable stage lengths in optimal control problems is introduced to enable accurate switching. In Chapter 10, I consider the optimal control of singular control problems that are very difficult to solve by other methods. In Chapter 11, the application of penalty functions is illustrated for the optimal control of systems where there are state constraints present. The time optimal control problem is considered in Chapter 12, and, in Chapter 13, the optimal control of nonseparable problems is illustrated with two examples. Since sensitivity is such an important issue, I have discussed that aspect in some detail in Chapter 14. In Chapter 15, I consider some practical aspects of applying optimal control to physical systems in practice and outline some areas for further research.

To enable the reader to gain direct experience with the computations, I have given listings of typical computer programs in their entirety in the appendix. The computer programs make the logic discussed in the text easier to follow, and the programs may be used by the reader to actually run some cases. It is through this type of direct experience that one gains the most insight into the computational aspects. Throughout the book I have also given computation times for some runs to give the reader some idea of what to expect. Whether a particular problem takes a few seconds or a few hours to run is useful information for the user. I have not made any special effort to maximize the efficiency of the computer programs. This exercise is left for the reader.

I am grateful to Professor Rutherford Aris for suggesting that I write this book and for providing encouragement during the writing process. I am also grateful to Professor Árpad Pethö for organizing the annual workshops in Germany and Hungary to which he has invited me to present the continuing developments of IDP. My thanks also go to the Natural Sciences and Engineering Council of Canada for supporting some of this work.

Rein Luus

Notation

a_{ij}	element of the i^{th} row and j^{th} column of $\mathbf{A}$ matrix
$\mathbf{A}$	state coefficient matrix $(n \times n)$
$\mathbf{B}$	control coefficient matrix $(n \times m)$
c	constant
c_i	cost associated with i^{th} job
$\mathbf{D}$	diagonal matrix of random numbers between -1 and 1
f	general function
f_i	i^{th} element of the vector f
$\mathbf{f}$	general vector function
g_i	continuous function of state variables introduced for convenience
h	height
h_i	i^{th} equality constraint
H	Hamiltonian
I	performance index
$\mathbf{I}$	identity matrix
J	augmented performance index
J_i	i^{th} job
L	length of a time stage
m	number of control variables
M	number of allowable values for each control variable chosen from uniform grid
n	number of state variables
N	number of grid points
P	number of time stages
q	pass number; raffinate solvent flow rate
Q	sum of squares of deviation
$\mathbf{Q}$	weighting matrix $(n \times n)$
$\mathbf{r}$	region vector over which allowable values of variables are chosen
R	number of randomly chosen values for control
$\mathbf{R}$	weighting matrix $(m \times m)$
s	shifting term
s_i	shifting term corresponding to constraint i
S	sum of absolute values
t	time
t_f	final time of operation
u	scalar control
u_j	j^{th} element of control vector $\mathbf{u}$
$\mathbf{u}$	control vector $(m \times 1)$
v	variable stage length; velocity

x_i	i^{th} state variable
$\mathbf{x}$	state vector $(n \times 1)$
z_i	i^{th} adjoint variable
$\mathbf{z}$	adjoint vector $(n \times 1)$

Greek letters

α	operator; positive constant
α_j	lower bound on control variable u_j
β	constant
β_j	upper bound on the control variable u_j
γ	region contraction factor by which the region is reduced after every iteration
δ	a small perturbation
ϵ	tolerance
η	region restoration factor
θ	penalty function factor
$\mathbf{\Theta}$	matrix $(n \times n)$
ρ	penalty function factor
τ	delay time
τ_i	time to execute job i
ϕ	integrand of performance index
Φ	final value performance index
$\mathbf{\Phi}$	transition matrix
$\mathbf{\Psi}$	matrix $(n \times m)$

Subscripts

$_{f}$	final time
$_{fc}$	calculated final time
$_{i}$	index
$_{in}$	initial value
$_{j}$	index
$_{k}$	index
$_{new}$	new value
$_{old}$	previous value
$_{p}$	predicted

Superscripts

*	best value obtained from previous iteration
^{d}	desired value
^{j}	iteration step
^{0}	optimal value
$^{(0)}$	initial value
^{q}	pass number
^{T}	transpose