

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

ITERATIVE

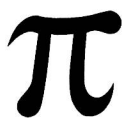
DYNAMIC

PROGRAMMING

REIN LUUS

CHAPMAN & HALL/CRC

0224
L975



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

110

ITERATIVE

DYNAMIC

PROGRAMMING

REIN LUUS



E200100579

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) I. Title. II. Chapman & Hall/CRC monographs and surveys in pure and applied mathematics; 110.

QA402.5 .L88 2000

519.7'03--dc21

99-058886

CIP

This book contains information obtained from authentic and highly regarded sources. Reprinted material is quoted with permission, and sources are indicated. A wide variety of references are listed. Reasonable efforts have been made to publish reliable data and information, but the author and the publisher cannot assume responsibility for the validity of all materials or for the consequences of their use.

Neither this book nor any part may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, microfilming, and recording, or by any information storage or retrieval system, without prior permission in writing from the publisher.

The consent of CRC Press LLC does not extend to copying for general distribution, for promotion, for creating new works, or for resale. Specific permission must be obtained in writing from CRC Press LLC for such copying.

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

Trademark Notice: Product or corporate names may be trademarks or registered trademarks, and are used only for identification and explanation, without intent to infringe.

© 2000 by Chapman & Hall/CRC

No claim to original U.S. Government works

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



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

110

ITERATIVE

DYNAMIC

PROGRAMMING

REIN LUUS

CHAPMAN & HALL/CRC

Monographs and Surveys in Pure and Applied Mathematics

Main Editors

H. Brezis, *Université de Paris*

R.G. Douglas, *Texas A&M University*

A. Jeffrey, *University of Newcastle upon Tyne (Founding Editor)*

Editorial Board

H. Amann, *University of Zürich*

R. Aris, *University of Minnesota*

G.I. Barenblatt, *University of Cambridge*

H. Begehr, *Freie Universität Berlin*

P. Bullen, *University of British Columbia*

R.J. Elliott, *University of Alberta*

R.P. Gilbert, *University of Delaware*

R. Glowinski, *University of Houston*

D. Jerison, *Massachusetts Institute of Technology*

K. Kirchgässner, *Universität Stuttgart*

B. Lawson, *State University of New York*

B. Moodie, *University of Alberta*

S. Mori, *Kyoto University*

L.E. Payne, *Cornell University*

D.B. Pearson, *University of Hull*

I. Raeburn, *University of Newcastle*

G.F. Roach, *University of Strathclyde*

I. Stakgold, *University of Delaware*

W.A. Strauss, *Brown University*

J. van der Hoek, *University of Adelaide*

This book is dedicated to
Professor Rutherford Aris

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 Árpád Pethő 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 A matrix
A	state coefficient matrix ($n \times n$)
B	control coefficient matrix ($n \times m$)
c	constant
c_i	cost associated with i^{th} job
D	diagonal matrix of random numbers between -1 and 1
f	general function
f_i	i^{th} element of the vector f
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
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
Q	weighting matrix ($n \times n$)
r	region vector over which allowable values of variables are chosen
R	number of randomly chosen values for control
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 u
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
Θ	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
Φ	transition matrix
Ψ	matrix ($n \times m$)

Subscripts

f	final time
f_c	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

Contents

1	Fundamental concepts	1
1.1	Introduction	1
1.2	Fundamental definitions and notation	2
1.2.1	Operator	2
1.2.2	Vectors and matrices	3
1.2.3	Differentiation of a vector	5
1.2.4	Taylor series expansion	5
1.2.5	Norm of a vector	6
1.2.6	Sign definite	6
1.2.7	Stationary and maxima (minima) points	7
1.3	Steady-state system model	7
1.4	Continuous-time system model	7
1.5	Discrete-time system model	8
1.6	The performance index	10
1.7	Interpretation of results	11
1.8	Examples of systems for optimal control	11
1.8.1	Linear gas absorber	11
1.8.2	Nonlinear continuous stirred tank reactor	13
1.8.3	Photochemical reaction in CSTR	15
1.8.4	Production of secreted protein in a fed-batch reactor	16
1.9	Solving algebraic equations	17
1.9.1	Separation of the equations into two groups	18
1.9.2	Numerical examples	19
1.9.3	Application to multicomponent distillation	31
1.10	Solving ordinary differential equations	32
1.11	References	32
2	Steady-state optimization	35
2.1	Introduction	35
2.2	Linear programming	35
2.2.1	Example – diet problem with 5 foods	38
2.2.2	Interpretation of shadow prices	42

2.3	LJ optimization procedure	44
2.3.1	Determination of region size	46
2.3.2	Simple example — 5 food diet problem	48
2.3.3	Model reduction example	48
2.3.4	Parameter estimation	54
2.3.5	Handling equality constraints	58
2.4	References	64
3	Dynamic programming	67
3.1	Introduction	67
3.2	Examples	67
3.2.1	A simple optimal path problem	68
3.2.2	Job allocation problem	69
3.2.3	The stone problem	72
3.2.4	Simple optimal control problem	73
3.2.5	Linear optimal control problem	75
3.2.6	Cross-current extraction system	76
3.3	Limitations of dynamic programming	80
3.4	References	80
4	Iterative dynamic programming	81
4.1	Introduction	81
4.2	Construction of time stages	82
4.3	Construction of grid for \mathbf{x}	82
4.4	Allowable values for control	82
4.5	First iteration	82
4.5.1	Stage P	83
4.5.2	Stage $P - 1$	83
4.5.3	Continuation in backward direction	83
4.6	Iterations with systematic reduction in region size	84
4.7	Example	85
4.8	Use of accessible states as grid points	85
4.9	Algorithm for IDP	86
4.10	Early applications of IDP	89
4.11	References	90
5	Allowable values for control	91
5.1	Introduction	91
5.2	Comparison of uniform distribution to random choice	92
5.2.1	Uniform distribution	93
5.2.2	Random choice	94
5.3	References	98

6	Evaluation of parameters in IDP	99
6.1	Introduction	99
6.2	Number of grid points	100
6.2.1	Bifunctional catalyst blend optimization problem	100
6.2.2	Photochemical CSTR	104
6.3	Multi-pass approach	106
6.3.1	Nonlinear two-stage CSTR system	107
6.4	Further example	109
6.4.1	Effect of region restoration factor η	111
6.4.2	Effect of the region contraction factor γ	112
6.4.3	Effect of the number of time stages	112
6.5	References	117
7	Piecewise linear control	119
7.1	Introduction	119
7.2	Problem formulation	119
7.3	Algorithm for IDP for piecewise linear control	120
7.4	Numerical examples	122
7.4.1	Nonlinear CSTR	122
7.4.2	Nondifferentiable system	124
7.4.3	Linear system with quadratic performance index	126
7.4.4	Gas absorber with a large number of plates	136
7.5	References	138
8	Time-delay systems	139
8.1	Introduction	139
8.2	Problem formulation	140
8.3	Examples	140
8.3.1	Example 1	140
8.3.2	Example 2	142
8.3.3	Example 3 – Nonlinear two-stage CSTR system	143
8.4	References	148
9	Variable stage lengths	149
9.1	Introduction	149
9.2	Variable stage-lengths when final time is free	155
9.2.1	IDP algorithm	156
9.3	Problems where final time is not specified	157
9.3.1	Oil shale pyrolysis problem	158
9.3.2	Modified Denbigh reaction scheme	161
9.4	Systems with specified final time	165
9.4.1	Fed-batch reactor	168
9.5	References	175

10 Singular control problems	177
10.1 Introduction	177
10.2 Four simple-looking examples	178
10.2.1 Example 1	178
10.2.2 Example 2	181
10.2.3 Example 3	187
10.2.4 Example 4	188
10.3 Yeo's singular control problem	191
10.4 Nonlinear two-stage CSTR problem	193
10.5 References	197
11 State constraints	199
11.1 Introduction	199
11.2 Final state constraints	199
11.2.1 Problem formulation	199
11.2.2 Quadratic penalty function with shifting terms	200
11.2.3 Absolute value penalty function	218
11.2.4 Remarks on the choice of penalty functions	223
11.3 State inequality constraints	224
11.3.1 Problem formulation	224
11.3.2 State constraint variables	225
11.4 References	234
12 Time optimal control	237
12.1 Introduction	237
12.2 Time optimal control problem	237
12.3 Direct approach to time optimal control	238
12.4 Examples	239
12.4.1 Example 1: Bridge crane system	239
12.4.2 Example 2: Two-link robotic arm	241
12.4.3 Example 3: Drug displacement problem	243
12.4.4 Example 4: Two-stage CSTR system	246
12.4.5 Example 5	249
12.5 High dimensional systems	251
12.6 References	253
13 Nonseparable problems	255
13.1 Introduction	255
13.2 Problem formulation	255
13.3 Examples	256
13.3.1 Example 1 – Luus-Tassone problem	256
13.3.2 Example 2 – Li-Haimes problem	264
13.4 References	264

14 Sensitivity considerations	265
14.1 Introduction	265
14.2 Example: Lee-Ramirez bioreactor	267
14.2.1 Solution by IDP	268
14.3 References	272
15 Toward practical optimal control	273
15.1 Introduction	273
15.2 Optimal control of oil shale pyrolysis	274
15.3 Future directions	278
15.4 References	282
A Nonlinear algebraic equation solver	283
A.1 Program listing	283
A.2 Output of the program	285
B Listing of linear programming program	287
B.1 Main program for the diet problem	287
B.2 Input subroutine	288
B.3 Subroutine for maximization	289
B.4 Output subroutine	290
C LJ optimization programs	291
C.1 Five food diet problem	291
C.2 Model reduction problem	293
C.3 Geometric problem	295
D Iterative dynamic programming programs	297
D.1 CSTR with piecewise constant control	297
D.2 IDP program for piecewise linear control	303
D.3 IDP program for variable stage lengths	308
E Listing of DVERK	315
E.1 DVERK	315
Index	321

Chapter 1

Fundamental concepts

1.1 Introduction

Optimization, or optimal control, in the sense to be used in this book, is concerned with determining the largest value or the smallest value for some criterion of performance. For example, if we are dealing with economic benefit, then we would like to choose the conditions for operating the system so that the economic benefit would be maximized. If, however, the criterion of performance is chosen to be the cost, then the system should be operated to minimize the cost. In each case we seek the operating conditions that yield the extreme value for the performance criterion.

It is obvious that the operating procedure is dictated by the choice of the criterion of operation. The choice of such criterion is not straightforward, since there are numerous factors that must be taken into consideration, such as productivity, profit, cost, environmental impact, reliability, yield of a reactor, quality of product, etc. We may want to have more than one criterion for optimization. For the present work, however, we assume that all the objectives can be expressed in terms of an appropriate scalar criterion of performance which we call *performance index*, with the understanding that the optimization results will be dependent on such a choice. It is also important to express this performance index in terms of the same variables that are used in the mathematical model of the physical system or process under consideration.

For the development of the *mathematical model* of the system, we need some insight into the behavior of the physical system, and how the variables at our disposal may be used to change its behavior. Such a relationship may be expressed in terms of algebraic equations, ordinary differential equations, difference equations, partial differential equations, integral equations, or combinations of them. The simplest situation arises, of course, if the model is described in terms of algebraic equations only. In this case we have a *steady-state optimization* problem, or we may simply call the process of finding the extreme value of the performance index *optimization*. If, however, the model consists of differential equations, difference equations, or integral