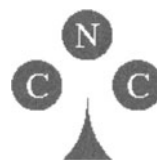


Natural Computing Series



Series Editors: G. Rozenberg

Th. Bäck A.E. Eiben J.N. Kok H.P. Spaink

Leiden Center for Natural Computing

Advisory Board: S. Amari G. Brassard M. Conrad

K.A. De Jong C.C.A.M. Gielen T. Head L. Kari

L. Landweber T. Martinetz Z. Michalewicz M.C. Mozer

E. Oja G. Păun J. Reif H. Rubin A. Salomaa M. Schoenauer

H.-P. Schwefel C. Torras D. Whitley E. Winfree J.M. Zurada

Springer-Verlag Berlin Heidelberg GmbH

A.E. Eiben • J.E. Smith

Introduction to Evolutionary Computing

With 73 Figures and 29 Tables



Springer

Prof. Dr. A.E. Eiben
Faculty of Sciences
Vrije Universiteit Amsterdam
De Boelelaan 1081a
1081 HV Amsterdam
The Netherlands
gusz@cs.vu.nl

Dr. J.E. Smith
Faculty of Computing,
Engineering
and Mathematical Sciences
University of the West of England
Bristol, BS16 1QY
U.K.

Series Editors

G. Rozenberg (Managing Editor)
rozenber@liacs.nl
Th. Bäck, J.N. Kok, H.P. Spaink
Leiden Center for Natural Computing
Leiden University
Niels Bohrweg 1
2333 CA Leiden, The Netherlands

A.E. Eiben
Vrije Universiteit Amsterdam

Library of Congress Cataloging-in-Publication Data

Eiben, Agoston E.

Introduction to evolutionary computing/A.E. Eiben, J.E. Smith.

p. cm. – (Natural computing series)

Includes bibliographical references and index.

1. Evolutionary programming (Computer science) 2. Evolutionary computation.
I. Smith, J. E. (James E.), 1964- II. Title. III. Series.

QA76.618.E33 2003

006.3–dc22

2003057316

ACM Computing Classification (1998): I.2, I.2.8, I.2.6

ISBN 978-3-642-07285-7 ISBN 978-3-662-05094-1 (eBook)

DOI 10.1007/978-3-662-05094-1

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilm or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer-Verlag. Violations are liable for prosecution under the German Copyright Law.

<http://www.springer.de>

© Springer-Verlag Berlin Heidelberg 2003

Originally published by Springer-Verlag Berlin Heidelberg New York in 2003.

Softcover reprint of the hardcover 1st edition 2003

The use of general descriptive names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

Cover Design: KünkelLopka, Werbeagentur, Heidelberg

Typesetting: Camera ready by the authors

45/3111 – 5 4 3 2 1 – Printed on acid-free paper

SPIN 11531296

GE: To my beloved wife Daphne

JS: To Cally and Jasper, constant sources of inspiration

Preface

This book is primarily a textbook for lecturers and graduate and undergraduate students. To this group the book offers a thorough introduction to evolutionary computing (EC), including the basics of all traditional variants (evolution strategies, evolutionary programming, genetic algorithms, and genetic programming); EC themes of general interest (such as algorithm parameter control, or constraint handling); a collection of particular EC techniques (e.g., niching, or coevolution); and an outlook to related areas (evolutionary art). This book is also meant for those who wish to apply EC to a particular problem or within a given application area. To this group the book is valuable because it presents EC as something to be *used*, rather than just being studied, and it contains an explicit treatment of guidelines for good experimentation. Last, but not least, this book contains information on the current state of the art in a wide range of subjects that are interesting to fellow researchers as quick reference on subjects outside of their own specialist field of evolutionary computing.

The motivation behind the book is education oriented. Both authors have many years of teaching experience, that is, have taught EC many times, not only within the context of a university, but also at EC-related summer schools for doctoral students, and at commercial courses for practitioners from business and industry. The lack of one good textbook that covers all necessary aspects of EC, contains the factual knowledge but also paying attention to the skills needed to use this technology has been repeatedly experienced. This resulted in a joint effort to fill this gap and produce the textbook both authors felt was missing. The educative role of the book is emphasised by the following features:

1. There are example applications in each chapter, except the chapter on theory.
2. Each chapter closes with exercises and a list of recommended further reading.
3. The book has a supporting a Web site with identical copies at:

- www.cs.vu.nl/~gusz/ecbook/ecbook.html
 - www.cems.uwe.ac.uk/~jsmith/ecbook/ecbook.html
4. On this Web site we outline a full academic course based on this material.
 5. There are slides to each chapter on the Web in PowerPoint and in PDF format. These slides can be freely downloaded and used to teach the material covered in the book.
 6. All illustrations used on the slides are available separately from the Web as source (editable), PostScript, and JPG files. These enable readers to use and reuse them and to create their own versions for their own slides.
 7. Furthermore, the Web site offers more exercises, answers to the exercises, downloadables for easy experimentation, errata, and a discussion group.

Writing this book would not have been possible without the support of many. In the first place, we wish to express our gratitude to Daphne and Cally for their patience, understanding, and tolerance. Without their support this book could not have been written. Furthermore, we acknowledge the help of our colleagues within EvoNet and the EC community. We are especially grateful to Larry Bull, Maarten Keijzer, Nat Krasnogor, Ben Paechter, Günter Raidl, Rob Smith, and Dirk Thierens for their comments on earlier versions of this book. The people in our departments also deserve a word of thanks for their support. Finally, Gusz Eiben wishes to thank András Lörincz and the ELTE University in Budapest for providing the facilities needed to finalise the camera ready copy during his stay in Hungary.

We wish everybody a pleasant and fruitful time reading and using this book.

Amsterdam, Bristol, Budapest, July 2003

Gusz Eiben and Jim Smith

Contents

- 1 Introduction 1**
 - 1.1 Aims of this Chapter 1
 - 1.2 The Main Evolutionary Computing Metaphor 1
 - 1.3 Brief History 2
 - 1.4 The Inspiration from Biology 3
 - 1.4.1 Darwinian Evolution 3
 - 1.4.2 Genetics 5
 - 1.5 Evolutionary Computing: Why? 7
 - 1.6 Exercises 13
 - 1.7 Recommended Reading for this Chapter 13

- 2 What is an Evolutionary Algorithm? 15**
 - 2.1 Aims of this Chapter 15
 - 2.2 What is an Evolutionary Algorithm? 15
 - 2.3 Components of Evolutionary Algorithms 18
 - 2.3.1 Representation (Definition of Individuals) 18
 - 2.3.2 Evaluation Function (Fitness Function) 19
 - 2.3.3 Population 20
 - 2.3.4 Parent Selection Mechanism 20
 - 2.3.5 Variation Operators 21
 - 2.3.6 Survivor Selection Mechanism (Replacement) 22
 - 2.3.7 Initialisation 23
 - 2.3.8 Termination Condition 23
 - 2.4 Example Applications 24
 - 2.4.1 The Eight-Queens Problem 24
 - 2.4.2 The Knapsack Problem 27
 - 2.5 Working of an Evolutionary Algorithm 29
 - 2.6 Evolutionary Computing and Global Optimisation 32
 - 2.7 Exercises 35
 - 2.8 Recommended Reading for this Chapter 35

3	Genetic Algorithms	37
3.1	Aims of this Chapter	37
3.2	Introductory Example	37
3.3	Representation of Individuals	40
3.3.1	Binary Representations	40
3.3.2	Integer Representations	41
3.3.3	Real-Valued or Floating-Point Representation	41
3.3.4	Permutation Representations	41
3.4	Mutation	42
3.4.1	Mutation for Binary Representations	43
3.4.2	Mutation Operators for Integer Representations	43
3.4.3	Mutation Operators for Floating-Point Representations	44
3.4.4	Mutation Operators for Permutation Representations	45
3.5	Recombination	46
3.5.1	Recombination Operators for Binary Representations	47
3.5.2	Recombination Operators for Integer Representations	49
3.5.3	Recombination Operators for Floating-Point Representations	50
3.5.4	Recombination Operators for Permutation Representations	52
3.5.5	Multiparent Recombination	57
3.6	Population Models	57
3.7	Parent Selection	59
3.7.1	Fitness Proportional Selection	59
3.7.2	Ranking Selection	60
3.7.3	Implementing Selection Probabilities	61
3.7.4	Tournament Selection	63
3.8	Survivor Selection	64
3.8.1	Age-Based Replacement	65
3.8.2	Fitness-Based Replacement	65
3.9	Example Application: Solving a Job Shop Scheduling Problem	66
3.10	Exercises	68
3.11	Recommended Reading for this Chapter	69
4	Evolution Strategies	71
4.1	Aims of this Chapter	71
4.2	Introductory Example	71
4.3	Representation	73
4.4	Mutation	73
4.4.1	Uncorrelated Mutation with One Step Size	75
4.4.2	Uncorrelated Mutation with n Step Sizes	76
4.4.3	Correlated Mutations	78
4.5	Recombination	80
4.6	Parent Selection	81
4.7	Survivor Selection	81

4.8	Self-Adaptation	82
4.9	Example Applications	84
4.9.1	The Ackley Function	84
4.9.2	Subjective Evolution of Colour Mixes	85
4.10	Exercises	86
4.11	Recommended Reading for this Chapter	87
5	Evolutionary Programming	89
5.1	Aims of this Chapter	89
5.2	Introductory Example	89
5.2.1	Representation	92
5.2.2	Mutation	92
5.3	Recombination	94
5.4	Parent Selection	94
5.5	Survivor Selection	95
5.6	Example Application	95
5.6.1	The Ackley Function	95
5.6.2	Evolving Checkers Players	96
5.7	Exercises	97
5.8	Recommended Reading for this Chapter	98
6	Genetic Programming	101
6.1	Aims of this Chapter	101
6.2	Introductory Example	101
6.3	Representation	103
6.4	Mutation	106
6.5	Recombination	107
6.6	Parent Selection	109
6.7	Survivor Selection	109
6.8	Initialisation	109
6.9	Bloat in Genetic Programming	110
6.10	Problems Involving “Physical” Environments	110
6.11	Example Application: Symbolic Regression	112
6.12	Exercises	113
6.13	Recommended Reading for this Chapter	113
7	Learning Classifier Systems	115
7.1	Aims of this Chapter	115
7.2	Introductory Example	115
7.3	General Background	118
7.4	ZCS: A “Zeroth-Level” Classifier System	120
7.5	XCS	122
7.5.1	Motivation	122
7.5.2	Description	123
7.6	Extensions	124

7.7	Example Applications	126
7.7.1	Modelling Financial Market Traders	126
7.7.2	A Multistep Problem	127
7.8	Exercises	128
7.9	Recommended Reading for this Chapter	128
8	Parameter Control in Evolutionary Algorithms	129
8.1	Aims of this Chapter	129
8.2	Introduction	129
8.3	Examples of Changing Parameters	132
8.3.1	Changing the Mutation Step Size	132
8.3.2	Changing the Penalty Coefficients	134
8.3.3	Summary	136
8.4	Classification of Control Techniques	137
8.4.1	<i>What</i> is Changed?	137
8.4.2	<i>How</i> are Changes Made?	138
8.4.3	What <i>Evidence</i> Informs the Change?	139
8.4.4	What is the <i>Scope</i> of the Change?	140
8.4.5	Summary	141
8.5	Examples of Varying EA Parameters	142
8.5.1	Representation	142
8.5.2	Evaluation function	143
8.5.3	Mutation	144
8.5.4	Crossover	145
8.5.5	Selection	146
8.5.6	Population	147
8.5.7	Varying Several Parameters Simultaneously	148
8.6	Discussion	150
8.7	Exercises	150
8.8	Recommended Reading for this Chapter	151
9	Multimodal Problems and Spatial Distribution	153
9.1	Aims of this Chapter	153
9.2	Introduction: Multimodal Problems and the Need for Diversity	154
9.2.1	Multimodal Problems	154
9.2.2	Genetic Drift	155
9.2.3	Biological Motivations and Algorithmic Approaches	155
9.2.4	Algorithmic Versus Genetic Versus Solution Space	156
9.2.5	Summary	157
9.3	Implicit Measures	158
9.3.1	Multiple Populations in Tandem: Island Model EAs	158
9.3.2	Spatial Distribution Within One Population: Diffusion Model EAs	160
9.3.3	Automatic Speciation Using Mating Restrictions	161
9.4	Explicit Diversity Maintenance	162

9.4.1	Fitness Sharing	163
9.4.2	Crowding	164
9.5	Multiobjective Evolutionary Algorithms	164
9.5.1	Multiobjective Optimisation Problems	164
9.5.2	Dominance and Pareto Optimality	166
9.5.3	EA Approaches to Multiobjective Optimisation	167
9.6	Example Application: Distributed Coevolution of Job Shop Schedules	170
9.7	Exercises	171
9.8	Recommended Reading for this Chapter	172
10	Hybridisation with Other Techniques: Memetic Algorithms	173
10.1	Aims of this Chapter	173
10.2	Motivation for Hybridising EAs	173
10.3	A Brief Introduction to Local Search	175
10.3.1	Lamarckianism and the Baldwin Effect	177
10.4	Structure of a Memetic Algorithm	178
10.4.1	Heuristic or Intelligent Initialisation	178
10.4.2	Hybridisation Within Variation Operators: Intelligent Crossover and Mutation	180
10.4.3	Local Search Acting on the Output from Variation Operators	181
10.4.4	Hybridisation During Genotype to Phenotype Mapping	182
10.5	Design Issues for Memetic Algorithms	183
10.5.1	Preservation of Diversity	183
10.5.2	Choice of Operators	184
10.5.3	Use of Knowledge	185
10.6	Example Application: Multistage Memetic Timetabling	185
10.7	Exercises	187
10.8	Recommended Reading for this Chapter	188
11	Theory	189
11.1	Aims of this Chapter	189
11.2	Competing Hyperplanes in Binary Spaces: the Schema Theorem	190
11.2.1	What is a Schema?	190
11.2.2	Holland's Formulation for the SGA	190
11.2.3	Schema-Based Analysis of Variation Operators	192
11.2.4	Walsh Analysis and Deception	192
11.2.5	Criticisms and Recent Extensions of the Schema Theorem	194
11.2.6	Gene Linkage: Identifying and Recombining Building Blocks	195
11.3	Dynamical Systems	196
11.4	Markov Chain Analysis	197
11.5	Statistical Mechanics Approaches	199

11.6	Reductionist Approaches	200
11.7	Analysing EAs in Continuous Search Spaces	201
11.8	No Free Lunch Theorems	201
11.9	Exercises	202
11.10	Recommended Reading for this Chapter	203
12	Constraint Handling	205
12.1	Aims of this Chapter	205
12.2	Constrained Problems	205
12.2.1	Free Optimisation Problems	207
12.2.2	Constraint Satisfaction Problems	207
12.2.3	Constrained Optimisation Problems	208
12.3	Two Main Types of Constraint Handling	209
12.4	Ways to Handle Constraints in EAs	210
12.4.1	Penalty Functions	212
12.4.2	Repair Functions	214
12.4.3	Restricting Search to the Feasible Region	215
12.4.4	Decoder Functions	216
12.5	Application Example: Graph Three-Colouring	217
12.5.1	Indirect Approach	217
12.5.2	Mixed Mapping – Direct Approach	218
12.6	Exercises	219
12.7	Recommended Reading for this Chapter	219
13	Special Forms of Evolution	221
13.1	Aims of this Chapter	221
13.2	Coevolution	221
13.2.1	Cooperative coevolution	222
13.2.2	Competitive coevolution	224
13.2.3	Example Application: Coevolutionary Constraint Satisfaction	226
13.3	Interactive Evolution	226
13.3.1	Optimisation, Design, Exploration	228
13.3.2	Interactive Evolutionary Design and Art	228
13.3.3	Example Application: The Mondriaan Evolver	229
13.4	Nonstationary Function Optimisation	232
13.4.1	Algorithmic Approaches	233
13.4.2	Selection and Replacement Policies	235
13.4.3	Example Application: Time-Varying Knapsack Problem	238
13.5	Exercises	239
13.6	Recommended Reading for this Chapter	240

14	Working with Evolutionary Algorithms	241
14.1	Aims of this Chapter	241
14.2	What Do You Want an EA to Do?	241
14.3	Performance Measures	244
14.3.1	Different Performance Measures	245
14.3.2	Peak versus Average Performance	250
14.4	Test Problems for Experimental Comparisons	252
14.4.1	Using Predefined Problem Instances	253
14.4.2	Using Problem Instance Generators	254
14.4.3	Using Real-World Problems	256
14.5	Example Applications	256
14.5.1	Bad Practice	256
14.5.2	Better Practice	257
14.6	Exercises	258
14.7	Recommended Reading for this Chapter	258
15	Summary	259
15.1	What Is in This Book?	259
15.2	What Is Not in This Book?	263
A	Gray Coding	265
B	Test Functions	267
B.1	Unimodal Functions	267
B.1.1	OneMax	267
B.1.2	The Sphere Model	267
B.1.3	Royal Road Function	268
B.2	Multimodal Functions	268
B.2.1	Ackley Function	268
B.2.2	Schwefel's function	268
B.2.3	Generalized Rastrigin's function	268
B.2.4	Deb's Deceptive 4-Bit Function	269
B.3	Randomised Test Function Generators	269
B.3.1	Kauffman's NK Landscapes	269
B.3.2	NKC Landscapes	271
B.3.3	Random L-SAT	271
	References	273
	Index	295