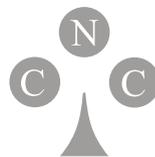


Natural Computing Series



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

A.E. Eiben • J.E. Smith

Introduction to Evolutionary Computing

Second Edition

 Springer

A.E. Eiben
Department of Computer Science
VU University Amsterdam
Amsterdam, The Netherlands

J.E. Smith
Department of Computer Science
and Creative Technologies
The University of the West of England
Bristol, UK

Series Editors

G. Rozenberg (Managing Editor)

Th. Bäck, J.N. Kok, H.P. Spaink
Leiden Center for Natural Computing
Leiden University
Leiden, The Netherlands

A.E. Eiben
VU University Amsterdam
The Netherlands

ISSN 1619-7127

Natural Computing Series

ISBN 978-3-662-44873-1

ISBN 978-3-662-44874-8 (eBook)

DOI 10.1007/978-3-662-44874-8

Library of Congress Control Number: 2015944812

Springer Heidelberg New York Dordrecht London

© Springer-Verlag Berlin Heidelberg 2003, 2015

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, 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.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made.

Printed on acid-free paper

Springer-Verlag GmbH Berlin Heidelberg is part of Springer Science+Business Media (www.springer.com)

Preface

This is the second edition of our 2003 book. It is primarily a book for lecturers and graduate and undergraduate students. To this group the book offers a thorough introduction to evolutionary computing (EC), descriptions of popular evolutionary algorithm (EA) variants, discussions of methodological issues and particular EC techniques. We end by presenting an outlook to evolutionary robotics and the future of EC, as it stands poised to make a major transition from evolution within computers to the evolution of things [147].

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 studied, and it contains an explicit treatment of guidelines for good experimentation. Finally, this book contains information on the 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 EC.

This book has a supporting website at

www.evolutionarycomputation.org

which offers additional information. In particular, the educational role of the book is emphasised:

1. There are exercises and a list of recommended further reading for each chapter.
2. The outline of a full academic course based on this book is given.
3. There are slides for each chapter in PDF and PowerPoint format. These slides can be freely downloaded, altered, and used to teach the material covered in the book.
4. Furthermore, the website offers answers to the exercises, downloadables for easy experimentation, a discussion forum, and errata.

When updating the book we altered its main logic. In the first edition, popular evolutionary algorithm variants, such as genetic algorithms or evolution strategies, had a prominent role. They were treated in separate chapters and

specific representations and evolutionary operators were presented within the framework of one of these algorithm variants. In the second edition we are emphasising the generic scheme of EAs as an approach to problem-solving. This is reflected by the following major changes:

- We added a chapter on problems. Since the whole book is about problem solvers, we felt it was good to start with a chapter on problems.
- The treatment of EAs is organised according to the main algorithm components, such as representation, variation and selection operators.
- The most popular EA variants are presented as special cases of the generic EA scheme. Although the treatment of each variant is now shorter, the list of variants is longer, now including differential evolution, particle swarm optimisation, and estimation of distribution algorithms.

We also extended the treatment of the how-to parts of the book. We added a new chapter on parameter tuning and grouped this with the chapters on parameter control and the how-to-work-with content into a methodological part. Furthermore, we dropped the Exercises and Recommended Reading sections at the end of each chapter as they were too static. Instead, we offer these on the website for the book.

The overall structure of the new edition is three-tier: Part I presents the basics, Part II is concerned with methodological issues, and Part III discusses advanced topics. These parts are followed by the References, and although that now contains nearly five hundred entries, we inevitably missed some. We apologise, it is nothing personal. Just send us an email if we forgot a really important one.

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 and the students worldwide who pointed out errors in and gave us feedback about the earlier version of the book. We are especially grateful to Bogdan Filipič for his comments on the almost-final draft of this book.

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

Amsterdam, Bristol, April 2015

Gusz Eiben and Jim Smith

Contents

Part I The Basics

1	Problems to Be Solved	1
1.1	Optimisation, Modelling, and Simulation Problems	1
1.1.1	Optimisation	2
1.1.2	Modelling	3
1.1.3	Simulation.....	5
1.2	Search Problems	5
1.3	Optimisation Versus Constraint Satisfaction	6
1.4	The Famous NP Problems	9
2	Evolutionary Computing: The Origins	13
2.1	The Main Evolutionary Computing Metaphor	13
2.2	Brief History	14
2.3	The Inspiration from Biology	15
2.3.1	Darwinian Evolution.....	15
2.3.2	Genetics	17
2.3.3	Putting It Together	19
2.4	Evolutionary Computing: Why?	20
3	What Is an Evolutionary Algorithm?	25
3.1	What Is an Evolutionary Algorithm?.....	25
3.2	Components of Evolutionary Algorithms	28
3.2.1	Representation (Definition of Individuals)	28
3.2.2	Evaluation Function (Fitness Function)	30
3.2.3	Population	30
3.2.4	Parent Selection Mechanism	31
3.2.5	Variation Operators (Mutation and Recombination) ...	31
3.2.6	Survivor Selection Mechanism (Replacement)	33
3.2.7	Initialisation	34
3.2.8	Termination Condition	34

3.3	An Evolutionary Cycle by Hand	34
3.4	Example Applications	36
3.4.1	The Eight-Queens Problem	36
3.4.2	The Knapsack Problem	39
3.5	The Operation of an Evolutionary Algorithm	41
3.6	Natural Versus Artificial Evolution	44
3.7	Evolutionary Computing, Global Optimisation, and Other Search Algorithms	46
4	Representation, Mutation, and Recombination	49
4.1	Representation and the Roles of Variation Operators	49
4.2	Binary Representation	51
4.2.1	Mutation for Binary Representation	52
4.2.2	Recombination for Binary Representation	52
4.3	Integer Representation	54
4.3.1	Mutation for Integer Representations	55
4.3.2	Recombination for Integer Representation	56
4.4	Real-Valued or Floating-Point Representation	56
4.4.1	Mutation for Real-Valued Representation	56
4.4.2	Self-adaptive Mutation for Real-Valued Representation	57
4.4.3	Recombination Operators for Real-Valued Representation	65
4.5	Permutation Representation	67
4.5.1	Mutation for Permutation Representation	69
4.5.2	Recombination for Permutation Representation	70
4.6	Tree Representation	75
4.6.1	Mutation for Tree Representation	77
4.6.2	Recombination for Tree Representation	78
5	Fitness, Selection, and Population Management	79
5.1	Population Management Models	79
5.2	Parent Selection	80
5.2.1	Fitness Proportional Selection	80
5.2.2	Ranking Selection	81
5.2.3	Implementing Selection Probabilities	83
5.2.4	Tournament Selection	84
5.2.5	Uniform Parent Selection	86
5.2.6	Overselection for Large Populations	86
5.3	Survivor Selection	87
5.3.1	Age-Based Replacement	88
5.3.2	Fitness-Based Replacement	88
5.4	Selection Pressure	90
5.5	Multimodal Problems, Selection, and the Need for Diversity	91
5.5.1	Multimodal Problems	91

5.5.2	Characterising Selection and Population Management	
	Approaches for Preserving Diversity	92
5.5.3	Fitness Sharing	92
5.5.4	Crowding	93
5.5.5	Automatic Speciation Using Mating Restrictions	95
5.5.6	Running Multiple Populations in Tandem: Island	
	Model EAs	95
5.5.7	Spatial Distribution Within One Population: Cellular	
	EAs	97
6	Popular Evolutionary Algorithm Variants	99
6.1	Genetic Algorithms	99
6.2	Evolution Strategies	101
6.3	Evolutionary Programming	103
6.4	Genetic Programming	104
6.5	Learning Classifier Systems	107
6.6	Differential Evolution	110
6.7	Particle Swarm Optimisation	112
6.8	Estimation of Distribution Algorithms	113

Part II Methodological Issues

7	Parameters and Parameter Tuning	119
7.1	Evolutionary Algorithm Parameters	119
7.2	EAs and EA Instances	120
7.3	Designing Evolutionary Algorithms	121
7.4	The Tuning Problem	123
7.5	Algorithm Quality: Performance and Robustness	125
7.6	Tuning Methods	128
8	Parameter Control	131
8.1	Introduction	131
8.2	Examples of Changing Parameters	132
	8.2.1 Changing the Mutation Step Size	133
	8.2.2 Changing the Penalty Coefficients	134
8.3	Classification of Control Techniques	136
	8.3.1 <i>What Is Changed?</i>	136
	8.3.2 <i>How Are Changes Made?</i>	136
	8.3.3 <i>What Evidence Informs the Change?</i>	138
	8.3.4 <i>What Is the Scope of the Change?</i>	138
	8.3.5 Summary	139
8.4	Examples of Varying EA Parameters	139
	8.4.1 Representation	139
	8.4.2 Evaluation Function	140

8.4.3	Mutation	141
8.4.4	Crossover	141
8.4.5	Selection	142
8.4.6	Population	142
8.4.7	Varying Several Parameters Simultaneously	143
8.5	Discussion	144
9	Working with Evolutionary Algorithms	147
9.1	What Do You Want an EA to Do?	147
9.2	Performance Measures	150
9.2.1	Different Performance Measures	151
9.2.2	Peak Versus Average Performance	155
9.3	Test Problems for Experimental Comparisons	158
9.3.1	Using Predefined Problem Instances	158
9.3.2	Using Problem Instance Generators	160
9.3.3	Using Real-World Problems	160
9.4	Example Applications	161
9.4.1	Bad Practice	161
9.4.2	Better Practice	162

Part III Advanced Topics

10	Hybridisation with Other Techniques: Memetic Algorithms	167
10.1	Motivation for Hybridising EAs	167
10.2	A Brief Introduction to Local Search	170
10.2.1	Lamarckianism and the Baldwin Effect	171
10.3	Structure of a Memetic Algorithm	172
10.3.1	Heuristic or Intelligent Initialisation	172
10.3.2	Hybridisation Within Variation Operators: Intelligent Crossover and Mutation	174
10.3.3	Local Search Acting on the Output from Variation Operators	175
10.3.4	Hybridisation During Genotype to Phenotype Mapping	176
10.4	Adaptive Memetic Algorithms	177
10.5	Design Issues for Memetic Algorithms	179
10.6	Example Application: Multistage Memetic Timetabling	181
11	Nonstationary and Noisy Function Optimisation	185
11.1	Characterisation of Nonstationary Problems	185
11.2	The Effect of Different Sources of Uncertainty	187
11.3	Algorithmic Approaches	189
11.3.1	Approaches That Increase Robustness or Reduce Noise	189
11.3.2	Pure Evolutionary Approaches to Dynamic Environments	189

- 11.3.3 Memory-Based Approaches for Switching or Cyclic Environments 190
- 11.3.4 Explicitly Increasing Diversity in Dynamic Environments 190
- 11.3.5 Preserving Diversity and Resampling: Modifying Selection and Replacement Policies 191
- 11.3.6 Example Application: Time-Varying Knapsack Problem 193
- 12 Multiobjective Evolutionary Algorithms 195**
 - 12.1 Multiobjective Optimisation Problems 195
 - 12.2 Dominance and Pareto Optimality 196
 - 12.3 EA Approaches to Multiobjective Optimisation 198
 - 12.3.1 Nonelitist Approaches 198
 - 12.3.2 Elitist Approaches 199
 - 12.3.3 Diversity Maintenance in MOEAs 199
 - 12.3.4 Decomposition-Based Approaches 200
 - 12.4 Example Application: Distributed Coevolution of Job Shop Schedules 200
- 13 Constraint Handling 203**
 - 13.1 Two Main Types of Constraint Handling 203
 - 13.2 Approaches to Handling Constraints 204
 - 13.2.1 Penalty Functions 206
 - 13.2.2 Repair Functions 208
 - 13.2.3 Restricting Search to the Feasible Region 209
 - 13.2.4 Decoder Functions 210
 - 13.3 Example Application: Graph Three-Colouring 211
- 14 Interactive Evolutionary Algorithms 215**
 - 14.1 Characteristics of Interactive Evolution 215
 - 14.1.1 The Effect of Time 216
 - 14.1.2 The Effect of Context: What Has Gone Before 216
 - 14.1.3 Advantages of IEAs 217
 - 14.2 Algorithmic Approaches to the Challenges of IEAs 217
 - 14.2.1 Interactive Selection and Population Size 217
 - 14.2.2 Interaction in the Variation Process 218
 - 14.2.3 Methods for Reducing the Frequency of User Interactions 218
 - 14.3 Interactive Evolution as Design vs. Optimisation 219
 - 14.4 Example Application: Automatic Elicitation of User Preferences 220
- 15 Coevolutionary Systems 223**
 - 15.1 Coevolution in Nature 223
 - 15.2 Cooperative Coevolution 224
 - 15.2.1 Partnering Strategies 225
 - 15.3 Competitive Coevolution 226

15.4	Summary of Algorithmic Adaptations for Context-Dependent Evaluation	227
15.5	Example Application: Coevolving Checkers Players	228
16	Theory	231
16.1	Competing Hyperplanes in Binary Spaces: The Schema Theorem	232
16.2	Criticisms and Recent Extensions of the Schema Theorem . . .	236
16.3	Gene Linkage: Identifying and Recombining Building Blocks . .	237
16.4	Dynamical Systems	238
16.5	Markov Chain Analysis	239
16.6	Statistical Mechanics Approaches	241
16.7	Reductionist Approaches	241
16.8	Black Box Analysis	242
16.9	Analysing EAs in Continuous Search Spaces	243
16.10	No Free Lunch Theorem	243
17	Evolutionary Robotics	245
17.1	What Is It All About?	245
17.2	Introductory Example	246
17.3	Offline and Online Evolution of Robots	248
17.4	Evolutionary Robotics: The Problems Are Different	250
17.5	Evolutionary Robotics: The Algorithms Are Different	253
17.6	A Glimpse into the Future	256
	References	259
	Index	283