Biologically Inspired Optimization Methods

Biologically Inspired Optimization Methods

An Introduction

M. Wahde
Chalmers University of Technology, Sweden

WIT*PRESS* **Southampton, Boston**

M. Wahde
Chalmers University of Technology, Sweden

Published by

WIT Press
Ashurst Lodge, Ashurst, Southampton, SO40 7AA, UK
Tel: 44 (0) 238 029 3223; Fax: 44 (0) 238 029 2853
E-Mail: witpress@witpress.com
http://www.witpress.com

For USA, Canada and Mexico

WIT Press
25 Bridge Street, Billerica, MA 01821, USA
Tel: 978 667 5841; Fax: 978 667 7582
E-Mail: infousa@witpress.com
http://www.witpress.com

British Library Cataloguing-in-Publication Data

A Catalogue record for this book is available
from the British Library

ISBN: 978-1-84564-148-1

Library of Congress Catalog Card Number: 2008924944

The texts of the papers in this volume were set individually by the authors or under their supervision.

Printed in Great Britain by Athenaeum Press Ltd.

For my parents

Contents

Preface

The advent of rapid, reliable and cheap computing power over the last decades has transformed many, if not most, fields of science and engineering. The multidisciplinary field of optimization is no exception. First of all, with fast computers, researchers and engineers can apply classical optimization methods to problems of larger and larger size. In addition, however, researchers have developed a host of new optimization algorithms that operate in a rather different way than the classical ones, and that allow practitioners to attack optimization problems where the classical methods are either not applicable or simply too costly (in terms of time and other resources) to apply.

This book is intended as a course book for introductory courses in stochastic optimization algorithms,[1] and it has grown from a set of lectures notes used in courses, taught by the author, at the international master programme Complex Adaptive Systems at Chalmers University of Technology in Göteborg, Sweden. Thus, a suitable audience for this book are third- and fourth-year engineering students, with a background in engineering mathematics (analysis, algebra and probability theory) as well as some knowledge of computer programming.

The organization of the book is as follows: first, Chapter 1 gives an introduction to the topic of optimization. Chapter 2 provides a brief background on the important (and large) topic of classical optimization. Chapters 3–5 cover the main topics of the book, namely stochastic optimization algorithms inspired by biological systems. Three such algorithms, or rather classes of algorithms as there are many different versions of each type, are presented: Chapter 3 covers evolutionary algorithms, Chapter 4 ant colony optimization and Chapter 5 particle swarm optimization. In addition to a presentation of the biological background of the algorithms, each of these chapters contains examples and exercises. Chapter 6 contains a performance study, comparing the various algorithms on a set of benchmark problems, thus allowing the student to select appropriate parameter settings for specific problems and to assess the advantages and weaknesses of each method. The book has four appendices, covering neural networks (Appendix A), an analysis of (some of) the properties of optimization algorithms (Appendix B), a brief background on data analysis (Appendix C) and a list of benchmark functions (Appendix D). Demoting

[1] In this book, the terms *optimization method* and *optimization algorithm* will be used interchangeably.

the entire topic of neural networks to an appendix may, perhaps, seem a bit unorthodox. Why not place neural networks on the same footing as the other algorithms? Well, the main reason is that neural networks, *per se*, do not constitute an *algorithm* but rather a *computational structure* to which several algorithms can be applied. There are many optimization algorithms specifically intended for neural networks (such as backpropagation, described in Appendix A), but it is also possible to apply the algorithms presented in Chapters 3–5 in order to optimize a neural network. Thus, in this book, rather than taking centre stage, neural networks (of which there are *many* different kinds!) form a backdrop. Another reason is the fact that, at Chalmers University of Technology, and many other universities, neural networks are taught as a separate topic. Thus, the placement (in this book) of neural networks in an appendix should certainly not imply that the topic is unimportant, but rather that its importance is such that it merits its own course.

At Chalmers University of Technology, courses are taught in quarters lasting 7 or 8 weeks. Assuming an 8-week quarter, a suggested schedule for a course based on this book could be as follows. Week 1: Introduction, classical optimization methods (Chapters 1–2); Week 2–5: Evolutionary algorithms, neural networks and data analysis (Chapter 3 and Appendices A and C); Week 6: Ant colony optimization (Chapter 4); Week 7: Particle swarm optimization (Chapter 5); Week 8: Comparison of algorithms (Chapter 6 and Appendix D). The contents of Appendix B can be included along the way, but can also be skipped altogether or just briefly considered, should the course be geared towards applications rather than theory.

Clearly, with the 8-week constraint just mentioned, it is not feasible to cover *all* stochastic optimization methods; hence, those sampled in this book represent a subset of the available methods, and one that is hopefully not too biased. Optimization algorithms that have been left out include tabu search, simulated annealing and reinforcement learning (even though, in a general sense, all stochastic optimization algorithms can be considered as versions of reinforcement learning). In addition, related topics such as cellular automata, fuzzy logic, artificial life and so on are not covered either. Also, in the topics that are considered, it has been necessary to leave out certain aspects. This is so, since there exist numerous versions of the stochastic optimization algorithms presented in Chapters 3–5. Thus, for example, while Chapter 3 considers genetic algorithms, (linear) genetic programming and interactive evolutionary computation, related algorithms such as evolution strategies and evolutionary programming are not discussed. Similarly, only two versions of ant colony optimization are considered in Chapter 4. In general, the presentation is centered on practical applications of the various algorithms. More philosophical topics, such as complexity, emergence, the relation between biological and artificial life forms and so on will not be considered.

Furthermore, regarding applications, it should be noted that multi-objective optimization, that is, problems in which the objective function (see Chapter 2) is represented as a vector rather than a scalar, and where, consequently, the notion of optimality is generally replaced by so-called Pareto optimality, will not be considered. However, even though non-scalar objective functions are excluded from the presentation, this does not prevent us from considering simultaneous optimization with respect to several, possibly conflicting, objectives since, at least in some problems such as the single-machine weighted tardiness problem considered in Chapter 4, a scalar objective function can be formed as a weighted sum of the functions representing the individual objectives.

Despite the limitations, it is the author's hope that this book will provide the reader with a suitable background for pursuing further studies of stochastic (and other) optimization algorithms. As a guide to such endeavours, this book is concluded with a bibliography for further reading.

Notation

Z denotes the set of integers. **R** denotes the set of real numbers, and $\mathbf{R}^n$ its n-dimensional equivalent,

$$\mathbf{R}^n = \{(x_1, \ldots, x_n) \, : \, x_i \in \mathbf{R}, \, i = 1, \ldots, n\}. \tag{N1}$$

Similarly $\mathbf{Z}^n$ denotes the n-dimensional equivalent of **Z**. The notation [a,b] is used to denote a closed interval in **R**, i.e. the set $\{x \, : \, a \leq x \leq b\}$. Similarly, $]a, b[$ denotes the open interval defined as $\{x \, : \, a < x < b\}$. As can be seen in eqn (N1), curly brackets are used for denoting sets in general. Curly brackets are also employed when listing a finite set of integers. For example $\{0, 1\}$ denotes the set consisting only of the two elements 0 and 1. The notation $A \subseteq B$ implies that A is a subset of B, meaning that every element of A is also an element of B. Vectors are written in bold lower-case characters, e.g. **x**. Here, **x** is to be understood as a column vector, i.e.

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}. \tag{N2}$$

To simplify the notation, however, a vector (or a point in $\mathbf{R}^n$) in component form is normally written $(x_1, x_2, \ldots, x_n)^{\mathrm{T}}$, where T denotes the transpose. Note that some lists of variables that are *not* vectors, strictly speaking, are written without the transpose; the chromosomes introduced in Chapter 3, which are sometimes written $c = (g_1, \ldots, g_m)$, where g_i denotes gene i, constitute an example. $\|\mathbf{x}\|$ denotes the **Euclidean norm** of a vector $\mathbf{x} \in \mathbf{R}^n$, i.e.

$$\|\mathbf{x}\| = \sqrt{\sum_{i=1}^{n} x_i^2}. \tag{N3}$$

$\|\mathbf{x} - \mathbf{y}\|$ denotes the (Euclidean) distance between two points **x** and **y** in $\mathbf{R}^n$.

The variables appearing in optimization problems are normally written x_i, $i = 1, \ldots n$, where n denotes the number of variables. An exception occurs in Appendix A, where x_i is used for denoting the input elements in neural networks. However, the abuse of notation is slight, since the inputs often (but not always) represent the variables of the problem, for example in cases where a neural network is used to fit a function $f(x_1, \ldots, x_n)$. In Chapter 4, where the problems considered involve searching for paths in a graph rather than optimizing a mathematical function $f(x_1, \ldots, x_n)$, n instead denotes the number of nodes in the graph. Furthermore, n is used in different ways in the application examples concluding Chapters 3–5.

The stochastic optimization algorithms presented in Chapters 3–5 are all population-based, i.e. they maintain a set of candidate solutions to the problem at hand. The number of elements in this set (referred to as the population size for genetic algorithms, see Chapter 3, or the swarm size for particle swarm optimization, see Chapter 5) is denoted as N. Stochastic optimization is normally carried out by a computer program implementing the algorithm in question. In this book, the execution of such an algorithm will be referred to as a **run**.

The letters i, j, are typically reserved for integer counters. In some cases, variables contain both subscripts and superscripts. In those cases where there is a risk of confusion, superscripts are put in brackets in order to distinguish them from exponents. Thus, $c^{[j]}$ denotes a variable c with a superscript j, whereas c^j denotes the j^{th} power of c. Some superscripts are, however, written without brackets, for example, x_{ij}^{pb} (Chapter 5), y_i^H (Appendix A), and $w_{ij}^{H \to O}$ (Appendix A). Since, throughout this book, exponents are always written using a single *lower-case* letter, there should be no risk of confusing the superscripts in the variables just listed with exponents.

Some algorithms involve iterations. In cases where there is sufficient space for a subscript or an argument (as in Chapter 2), the iterates are normally enumerated (e.g. as $\mathbf{x}_j$ or $x(j)$, whichever is most convenient for the application at hand). However, when, for example, a variable already has several subscripts, for instance, x_{ij}, new iterates are typically not enumerated explicitly. Instead, the next iterate is denoted as

$$x_{ij} \leftarrow x_{ij} + \Delta x_{ij}. \tag{N4}$$

A left-pointing arrow thus signifies that a new value is assigned to the variable shown to the left of the arrow. In addition, some elements of notation are only relevant to a particular chapter, and will therefore be described when introduced.

Whenever a new technical term is introduced and briefly described, it is written with **bold** letters. At the very end of the book, all technical terms are summarized in the form of an index.

Acknowledgements

I thank my family and friends, as well as colleagues and PhD students, for their patience and understanding during the writing of this book, and also for helping me with the proof-reading. I also express my gratitude to the many students who have suffered through early drafts of this book and have helped improve the book by finding misprints and other errors, most of which have hopefully been corrected. Any remaining errors are my own.

Furthermore, I would like to thank particularly K. S. Srikanth and the production team of Macmillan Publishing Solutions for their excellent typesetting. I am also grateful to Terri Barnett, Isabelle Strafford and Elizabeth Cherry at WIT Press, for their patience with my many delays. Last, but not least, I wish to thank Prof. Carlos Brebbia for inviting me to write this book in the first place.