Springer Series in Operations Research and Financial Engineering

Charles Audet
Warren Hare

# Derivative-Free and Blackbox Optimization

Springer

# Springer Series in Operations Research and Financial Engineering

More information about this series at http://www.springer.com/series/3182

Charles Audet • Warren Hare

# Derivative-Free and Blackbox Optimization

Springer

Charles Audet
Dépt. Mathématiques et Génie Industriel
Ecole Polytechnique de Montréal
Montréal, Québec, Canada

Warren Hare
Department of Mathematics
University of British Columbia
Kelowna, BC, Canada

ISSN 1431-8598 ISSN 2197-1773 (electronic)
Springer Series in Operations Research and Financial Engineering
ISBN 978-3-319-68912-8 ISBN 978-3-319-68913-5 (eBook)
https://doi.org/10.1007/978-3-319-68913-5

Library of Congress Control Number: 2017958449



Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer International Publishing AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

# Foreword

This timely book provides an introduction to blackbox optimization. This is often shortened to BBO, and it refers to an optimization problem in which some or all of the problem functions are gotten by running a computer subroutine with the optimization trial point in the input parameters. Some of these routines are very expensive to execute for a given argument. I was involved in a problem where the cost of a function value was three weeks on four SGI processors. Despite the expense, BBO is a topic of crucial importance to many applications. This is especially true for important design applications based on legacy simulation codes that do not lend themselves to automatic differentiation or the computation of adjoints.

I was told by an engineering executive that his company felt little incentive to commit scarce resources to redo these legacy codes to make it more likely for them to produce derivatives using adjoint solvers or automatic differentiation. His reasoning was since his department had not done nearly everything they could with the legacy codes he was not ready to pay for the reimplementation. This is unlikely to change anytime soon; research money is always tight. Thus, DFO algorithms, optimization algorithms that do not use derivatives to calculate steps, are unlikely to decrease in importance for applications. Besides, I have also been told by some engineers that they do not trust the optima found using gradient-based algorithms because of the belief, justified or not, that they are too dependent on the starting point of the gradient-based algorithm. There is some truth to this as you will see if you use a DFO algorithm that incorporates a more broadly based exploration of design space, like the MADS or surrogate-based algorithms covered in this book. Still, I do not want to overemphasize this point. I would lean towards using derivatives if I had them.

Another important aspect of BBO treated in the book is the various forms taken by the constraints in such problems. They may not be numerical in the sense that a given set of the constraints may just return a “yes” or a “no” telling whether the argument is feasible without a number quantifying the violation.

If you want your code to be used for more than academically staged beauty contests involving test problems with analytic or no constraints, then you had better be able to deal with nonnumerical constraints. These include the so-called hidden constraints for which infeasibility is indicated by not returning a meaningful value from the underlying simulation. I was involved in a project once for which the objective function value of 2.6 was often returned. It turned out that the programmer had decided to return 2.6 whenever the objective function subroutine failed to complete the function or constraint evaluation. Unfortunately, this was well within the range of meaningful function values and so we scratched our heads for an hour or so over some strange optimization steps. I was involved in another application for which hidden constraints were violated at about two-thirds of our trial points.

Let me be clear that artificial test problems may be difficult, and nonnumerical constraints may or may not be difficult to treat for a given problem, though obviously they do require different approaches than numerical constraints. My contention is that it is pointless to advertise a DFO routine as a BBO routine if it cannot treat nonnumerical constraints.

Efficiency is another issue that has a different flavor for BBO problems. It is nicer to take fewer expensive calls to the blackbox routine to get an optimum, but it is not all that counts. Reliability in finding an optimizer with not too many evaluations is also important. One industrial group decided to use MADS in their in-house software for an important application because it never took more than 15 minutes to find an answer for instances of a recurring problem. Their own excellent nonlinear interior point code sometimes took milliseconds to find an answer, but sometimes it failed. Fifteen minutes was acceptable, but failure was expensive because of equipment that had to wait until the problem was solved each time.

The hardest decision a textbook writer has to make is what to leave out. There are always topics trying to worm their way into the book. If you let them in, then you may find that you have written an encyclopedic volume that is expensive for the student to buy and a difficult book to teach from unless the teacher is an expert. Some optimization books are like this. I believe that Charles and Warren have made reasonable choices to avoid that trap.

Charles and Warren are an interesting team. Charles is a major research contributor to the subject with some truly amazing applications under his belt; my personal favorite is the placement of Québec snowfall sensors—lots of nonnumerical constraints.

Warren has come more lately to this area, and that gives him a fresher eye than Charles might have in knowing what topics are difficult for the novice and how much detail is needed to explain them. My spot reading in the book makes me think that this would be a fine text to teach from at the advanced undergraduate/beginning graduate level. It does not try to do too much or too little on any topic. I really enjoyed the lucid presentation in

Chap. 5 of the McKinnon example showing that pure Nelder-Mead can fail analytically—we all knew it could fail numerically. Of course, that does not stop its use, it just needs to be restarted with a fresh simplex when it fails. However, that is an inconvenience.

This book is explicit about the desire to make nonsmooth analysis a topic studied by all optimizers, and I hope it will accomplish this important goal. This book does a thorough job presenting the theory behind the methods treated here, and that requires nonsmooth analysis. I applaud the authors for including the theory as well as the practical details for the algorithms treated. It is more common to do one or the other.

It is always irritating to me when important researchers say, or imply, that theory is not important for methods intended to solve real problems. I could not disagree more. A theorem should say what happens when a method is applied to a problem when the hypotheses hold. It is a proof of correctness for the algorithm on the class of problems satisfying the hypotheses. It is not that the hypotheses should be checked before the algorithm is used. I will try to make my point with a simple example: the theory says that a square matrix is positive definite iff the Cholesky decomposition of that matrix exists. But, we do not check to see that the matrix is positive definite before attempting the decomposition. If the decomposition fails, then we know.

I firmly believe that if a method works well on a class of problems, then mathematics can give insight into why this happens. It is only that we are sometimes unable to discover the right mathematics. But we must keep trying. This book gives valuable background for the quest, which I hope will be taken up by those who study this book.

Noah Harding Professor Emeritus
Department of Computational
& Applied Mathematics
Rice University
Houston, TX, USA

John E. Dennis, Jr.

# Preface

This book is inspired by one goal and one belief. Our goal is to provide a clear grasp of the foundational concepts in derivative-free and blackbox optimization, in order to push these areas into the mainstream of nonlinear optimization. Our belief is that these foundational concepts have become sufficiently mature that it is now possible to teach them at a senior undergraduate level.

Derivative-free and blackbox optimization have made massive advances over the past two decades and, in our opinion, represent one of the most rapidly expanding fields of nonlinear optimization research. We also feel that derivative-free and blackbox optimization represent one of the most important areas in nonlinear optimization for solving future applications in real-world problems.

We target this book at two broad audiences and hope that both will find value.

Our first audience is individuals interested in entering (or better understanding) the fascinating world of derivative-free and blackbox optimization. We do not present the absolute state of the art in modern algorithms and theory, as we feel that it belongs in the realm of research papers. Instead we focus on the foundational material required to understand and appreciate the state of the art. To this end, in addition of studying several optimization methods, this book includes an elementary introduction to parts of nonsmooth analysis that have proved useful in conceiving and analyzing the methods given here. Nonsmooth analysis is not covered often enough outside the nonsmooth research community, and we hope our small contribution will help bridge this gap. The book also presents rigourous convergence theory for the algorithms in a way suitable for students in the mathematical sciences or in engineering. We hope that if more students see what can be done, both theoretically and practically, via derivative-free optimization, they may choose to make careers in this rewarding research area. Moreover, we hope this book will leave them well prepared to step into the world of derivative-free and blackbox optimization.

The second audience is practitioners who have real-world problems to solve that cannot be approached by traditional gradient-based methods. In the past, many such practitioners have fallen back on *ad hoc* methods, resulting in a plethora of papers publishing incremental improvements to solution quality. The methods covered in this book have proven convergence results, mathematically supported stopping criterion, and a track record of practical success. Yet, for all of this, the methods are nonetheless easy to use and elegant in their simplicity. While we do not feel that the methods herein are the only possible approaches to real-world problems, we hope they can provide better and more consistent starting points for future applications.

Finally, we remark that, for pedagogical purposes, we will present our algorithms not necessarily as they were discovered, but as hindsight makes us wish they had been.

**Some Special Thanks.** From Charles Audet's point of view, this book began through joint work with John Dennis. Certain portions of this book come from notes generated during past projects, and we are very grateful for these early interactions. From Warren Hare's point of view, this book began through the University of British Columbia supporting a sabbatical year, much of which was spent in the GERAD Research Center at the Université de Montréal Campus.

Both authors wish to express special thanks to Sébastien Le Digabel, who helped design the table of contents and only avoided further collaboration (on this project) by becoming a father for the second time. Thanks also to Catherine Poissant, who diligently ensured that all exercises were doable, and to Christophe Tribes and Yves Lucet, for helping with some of the numerical examples within this book. Finally, thanks to the students of the winter 2017 classes in Kelowna and in Montréal and to Chayne Planiden and Viviane Rochon Montplaisir, who proofread and made some precious and constructive comments on the book.

Special thanks to the Banff International Research Station (BIRS) for hosting us during the final stages of completing this book.

And, of course, a huge thank-you to our families – Annie, Xavier, Blanche, Émile, Shannon, and Alden – for supporting us through this project and understanding the importance of properly optimized octagonal whiskey glasses.

**Expected Background.** Readers of this textbook are expected to have an understanding of multivariate calculus (gradients, Hessians, multi-variable integration, etc.), linear algebra (vectors, matrices, determinants, linear independence, etc.), and mathematical proof techniques (contraposition, induction, contradiction, etc.). These subjects are typically taught in first or second year of mathematics, engineering, computer science, and other degrees focused in quantitative science. We recommend completing courses in each of these before reading this book.

**How to Use This Book.** This book is designed as a textbook, suitable for self-learning or for teaching an upper-year university course. We envision the book as suitable for either a one- or two-term course and have designed it to be modular, so it is not necessary to read cover to cover, from page 1 to page 300. Nonetheless, certain portions should be read before continuing through the book. The book is separated into five main parts, plus an appendix.

- **Part 1: Introduction and Background Material.** This part contains introductory material and the necessary background for reading the remainder of the book. Chapter 1 (Introduction: Tools and Challenges) introduces the ideas and purpose of derivative-free optimization (DFO) and blackbox optimization (BBO) and includes three motivational examples. Chapter 2 (Mathematical Background) provides definitions and notations used throughout the book, including brief primers on convexity and simplices. These two concepts are of great importance throughout the book, so we recommend that readers are comfortable with the material before moving further. Chapter 3 (The Beginnings of DFO Algorithms) introduces the first algorithms in the book. While the algorithms in Chapter 3 are naive and generally ineffective, they provide the building blocks for many other methods and their analyses. Therefore, while not strictly necessary, we recommend that readers study this chapter before moving further.
- **Part 2: Popular Heuristic Methods.** This part examines classical heuristics for solving optimization problems without using gradient evaluations. While these methods do not meet *our* definition of a DFO method, they are nonetheless popular and effective methods for solving optimization problems without using gradients. Chapter 4 (Genetic Algorithms) can be read as a stand-alone chapter, as can Chapter 5 (Nelder-Mead). Part 2 can be skipped entirely without affecting understanding of Part 3 or 4.
- **Part 3: Direct Search Methods.** This part focuses on the first of two major categories of DFO methods: direct search methods. In these methods, the DFO algorithm evaluates function values at a collection of points and acts based on those values without any derivative approximation. As such, the methods can be effective in BBO even when the objective and constraint functions are nonsmooth. These methods apply and expand on results from linear algebra and provide a great opportunity to delve into this subject. Chapter 6 (Positive Bases and Nonsmooth Optimization) provides some important background material for understanding direct search methods, including elements of the nonsmooth analysis. Chapter 7 (Generalised Pattern Search) explores a basic direct search method for unconstrained optimization and can be read after completing the first four sections of Chapter 6. Chapter 8 (Mesh

Adaptive Direct Search) advances from Chapter 7, so it should only be read after completing that chapter. The method proposed in Chapter 8 handles nonsmooth constraints and relies on the material from Section 6.5 for the convergence analysis. Part 3 can be skipped entirely without affecting understanding of Part 2 or 4.

- **Part 4: Model-Based Methods.** This part focuses on the second of two major categories of DFO methods: model-based methods. In these methods, the DFO algorithm uses function values to build an approximation model of the objective and uses the model to guide future iterations. These methods are effective when the objective function is expected to be smooth (although gradients are not analytically available). The techniques use numerical analysis to prove that classical gradient-based algorithms can be applied when gradients are approximated. Chapter 9 (Building Linear and Quadratic Models) is a stand-alone chapter that shows how to construct linear and quadratic models of the objective function. Chapter 9 is designed to answer a key question raised in Chapters 10 and 11. Chapters 10 (Generalised Model-Based Descent) and 11 (Model-Based Trust Region) are mostly stand-alone chapters that cover two popular frameworks for model-based DFO. However, both chapters use definitions from Section 9.1, so they should only be read after completing that section. Part 4 can be skipped entirely without affecting understanding of Part 2 or 3.
- **Part 5: Extensions and Refinements.** This part discusses how to develop and use BBO and DFO methods efficiently in practice. Chapter 12 (Variables and Constraints) discusses various types of variables and proposes a more subtle approach to handle relaxable constraints than the simple strategy of rejecting any infeasible point. Chapter 13 (Optimization Using Surrogates and Models) discusses a way to use surrogates and models of the objective function and constraints, which is crucial for improving efficiency. This chapter relies on Part 3 for the algorithmic mechanism and on Chapter 9 for building models. Chapter 14 (Biobjective Optimization) discusses situations where there is not one but two conflicting objective functions to be optimized. It shows how to approach this question by solving a series of single-objective reformulations. Each chapter of this part is independent and may be skipped.
- **Appendix: Comparing Optimization Methods.** This appendix provides a very brief overview of good practices when performing numerical tests to compare optimization algorithms. The part also includes a large project, which applies material from Parts 1, 2, 3, and 4 in a single problem.

In order to help the book be a suitable teaching aid, we have strived to make each chapter approximately equal to 3 hours of lecture and included a collection of exercises at the end of each chapter.

Exercises that require the use of a computer are tagged with a box symbol ($\square$) and the more difficult ones by a plus symbol (+). Some are tagged by both symbols.

For readers interested in the history of the subject, each part concludes with **Remarks** that provide references and historical context. These only provide some very high-level remarks and are by no means a complete historical overview. The Remark portions also include some comments on new results in the field and modern research directions.

Montréal, QC, Canada — Charles Audet
Kelowna, BC, Canada — Warren Hare

# *Contents*