Engineering Optimization

WILEY-ASME PRESS SERIES LIST

Engineering Optimization Applications, Methods and Analysis	Rhinehart	March 2018
Robust Adaptive Control for Fractional-Order Systems with Disturbance and Saturation	Chen	November 2017
Robot Manipulator Redundancy Resolution	Zhang	October 2017
Combined Cooling, Heating, and Power Systems: Modeling, Optimization, and Operation	Shi	August 2017
Applications of Mathematical Heat Transfer and Fluid Flow Models in Engineering and Medicine	Dorfman	February 2017
Bioprocessing Piping and Equipment Design: A Companion Guide for the ASME BPE Standard	Huitt	December 2016
Nonlinear Regression Modeling for Engineering Applications	Rhinehart	September 2016
Fundamentals of Mechanical Vibrations	Cai	May 2016
Introduction to Dynamics and Control of Mechanical Engineering Systems	To	March 2016

Engineering Optimization

Applications, Methods, and Analysis

R. Russell Rhinehart

This Work is a co-publication between ASME Press and John Wiley & Sons Ltd.

WILEY

ASME PRESS

This edition first published 2018

This Work is a co-publication between ASME Press and John Wiley & Sons Ltd

Registered Offices
John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, USA
John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex, PO19 8SQ, UK

Editorial Office
The Atrium, Southern Gate, Chichester, West Sussex, PO19 8SQ, UK

For details of our global editorial offices, customer services, and more information about Wiley products visit us at www.wiley.com.

Wiley also publishes its books in a variety of electronic formats and by print-on-demand. Some content that appears in standard print versions of this book may not be available in other formats.

Library of Congress Cataloging-in-Publication Data

Names: Rhinehart, R. Russell, 1946– author.
Title: Engineering optimization : applications, methods and analysis / by R. Russell Rhinehart.
Description: First edition. | Hoboken, NJ : John Wiley & Sons, 2018. | Includes index. |
Identifiers: LCCN 2017052555 (print) | LCCN 2017058922 (ebook) | ISBN 9781118936313 (pdf) | ISBN 9781118936320 (epub) | ISBN 9781118936337 (cloth)
Subjects: LCSH: Engineering–Mathematical models. | Mathematical optimization.
Classification: LCC TA342 (ebook) | LCC TA342 .R494 2018 (print) | DDC 620.001/5196–dc23
LC record available at https://lccn.loc.gov/2017052555

Cover design: Wiley
Cover image: © John Wiley & Sons

Set in 10/12pt Warnock by SPi Global, Pondicherry, India

Printed and bound by CPI Group (UK) Ltd, Croydon, CR0 4YY

10 9 8 7 6 5 4 3 2 1

Contents

Preface

Introduction

Optimization means seeking the best outcome or solution. It is an essential component of all human activities. Whether personal or professional, we seek best designs, best choices, best operation, more bang for the buck, and continuous improvement.

Here are some professional examples: Minimize work events that lead to injury while remaining economically competitive. Structure workflow to maximize return on investment. Design an antenna that maximizes signal clarity for a given power. Define a rocket thrust sequence to maximize height. Determine the number of parallel devices to minimize initial cost plus future risk.

Here are some personal examples: Seek the best vacation experience for the lowest cost. Minimize grocery bill, but meet desires for nourishment and joy of eating. Set the family structure for raising children that leads to well-adjusted, happy, productive outcomes, but keep within the limits of personal resources. Create a workout regime that leads to fastest and most attractive muscle development, with no injury, and in balance with other desires in quality of life.

Optimization is not just an intellectual exercise; although often, solving the challenge is as rewarding as completing a Sudoku puzzle. We implement the optimized decision. Accordingly, within any application it is essential to completely and appropriately assess the metrics that quantify "best." If the description of what you want to achieve is not quite right, then the answer will also be wrong, which the implementation will reveal in retrospect. You want to get it right prior to implementation. So, part of this book is about development of the optimization objective.

After the objective is stated, we desire an efficient search logic to find the best solution, with precision and with minimal computational and experimental effort. So, other parts of this book are about the optimizer—the search logic, or algorithm.

Both aspects are essential, and I find that most books on optimization focus on the intellectually stimulating mathematics of the algorithms. So, I offer this book to provide a balance of essential topics to the application to guide user choices in structuring the objective, defining constraints, choosing convergence, choosing initialization, etc. Some will be disappointed that this book is not a compendium of every optimization algorithm conceived by mankind. However, others will value the application perspective.

Also, I find that most people using optimization as a tool did not have a course on it while in school. So, I have written this book in a style that I hope facilitates self-study by those who need to understand optimization applications while keeping it fit for use as a graduate-school course textbook.

Key Points

Here are a few essential aspects of optimization:

Point 1: Although optimization offers the joys of solving an intellectual puzzle, it is not just a stimulating mathematical game. Optimization applications are complicated, and the major challenges are the clear and complete statement of:

1) The objective function (OF—the outcome you wish to minimize or maximize)
2) Constraints (what cannot or should not be violated, or exceeded)
3) The decision variables (DV—what you are free to change to seek a minimum)
4) The model (how DVs relate to OF and constraints)
5) The convergence criterion (the indicator of whether the algorithm has found a close enough proximity to the minimum or maximum and can stop or needs to continue)
6) The DV initialization values
7) The number of starts from randomized locations to be confident that the global optimum has been found
8) The appropriate optimization algorithm (for the function aberrations, for utility, for precision, for efficiency)
9) Computer implementation in code   Oh yes,
10) The mathematics of the optimization algorithm (understanding this is also important)

This book seeks to address all 10 aspects, not just the 10th.

Point 2: Do not study. Learning is most effective if you integrate the techniques into your daily life. You will forget the material that you memorized in order to pass a test. Since this book provides skills that are essential for both personal and career life, I want you to take the techniques with you. I want this book to be useful in your future. Although memorization and high-level mathematical analysis are both elements of the book, understanding the examples and doing of the exercises is more important. To maximize the impact of this material, you need to integrate it into your daily life. You need to practice it.

Oh, I see I omitted a comma in the first sentence of the paragraph above. It should have been “Do, not study.” Learn by doing. After you read a section and think you understand it, see if you can implement it. Of course, the comma “error” above was intentional to wake up curiosity about the message.

Point 3: Optimization is universal to all engineering, business, science, computer science, and technology disciplines. Although primarily written for engineering applications, this introductory book is designed to be useful for all those seeking to apply optimization in all fields.

Point 4: The implementation of optimization requires computer programming, which for many is an aggravation. To help the reader, I currently have, and plan to support, a website that offers to any visitor optimization software and examples. Visit www.r3eda.com. The “r3” in the address is my initials, and the appended “eda” means “enabling data analysis.” Seeking to maximize ease of use and accessibility, the programs are written as VBA macros for MS Excel. VBA is not the fastest-computing environment, nor does it have the best scientific data processing functions. However, it has been adequately functional for all of my applications, and if you need something better, the code can be translated. This book provides a VBA primer (Appendix F) for those needing the help in accessing and modifying the code. The programs on the r3eda site solve many of the examples in this book.

Book Aspirations

Readers should be pleased with their ability to:

- Understand and use the fundamental mathematical techniques associated with optimization
- Define objective functions, decision variables, models, and constraints for a variety of optimization applications
- Develop, modify, and program simplified versions of the more common optimization algorithms
- Understand and choose appropriate methods for:
  - Constrained optimization
  - Global optimization
  - Convergence criteria
  - Surface aberrations
  - Stochastic applications
- Understand diverse issues related to optimizer desirability
- Explore, contrast, and evaluate the performance of optimization algorithms and user choices of convergence criteria, numerical derivative estimation, threshold, constraint handling method, parameter values, etc. with respect to precision, user convenience, and other measures of optimizer desirability
- Apply optimization algorithms to case studies relevant to the reader's career
- Continue learning optimization methods from texts, reports, Internet postings, and refereed journal articles

Optimization is the name for the procedure for finding the best choices. "Procedure," "best," and "choices" are separate aspects, and the user must understand each to be able to appropriately define the application. And each aspect has a large range of options.

Procedure

This relates to the method used to find the optimum:

- In process or device design, for example, the choices could be the equipment specifications (type, materials, size), and the evaluation of best in the design could be to minimize capital cost with a constraint on reliability. With mixed continuous, discrete, and class variables as the choices, a direct search algorithm might be the best optimizer.
- Alternately, in scheduling a rocket thrust to reach a desired height, the stage choices might be height, best might be evaluated as minimizing either time or fuel use, and the appropriate algorithm might be dynamic programming.
- Another example is characterized as the traveling salesman problem in which the objective is to determine a sequence of locations to visit to minimize travel distance. Here the choice is the sequence, and the best sequence might be impacted by a priority of visits, expenses, wasted time, etc. The procedure might use the random keys method to convert a sorted list of rational numbers into the sequence.
- As a final contrasting example, in model-predictive control, the objective might be to minimize time to move a response to a set point while penalizing excessive manipulated variable moves while avoiding constraints; and the choices might be the future sequence of manipulations. If the penalties are quadratic, the appropriate algorithm might be a gradient-based procedure.

Best

Within optimization terminology, the definition of best for a specific application (and the method for calculating a value to quantify best) is variously termed the cost function or the objective function (OF). It is the function that returns a value representing an assessment of goodness. Best usually means minimize undesirable aspects and/or maximize desirable aspects, and the OF can represent a wide range of metrics related to economics, safety, time, resource conservation, quality, deviation, probability, etc. But best might mean to minimize a worst-case feature (min the max, or min–max), such as finding a path through mountains that minimizes the steepest ascent or finding a process design that minimizes the worst-case outcome (risk).

Defining the appropriate OF is situation specific, and often it is the key challenge in an optimization application. The user needs to clearly understand the complex situation and realize that a first statement of the OF usually embodies a superficial understanding. Subsequent analysis of the results will lead to an evolution of the OF. For example, a challenge might be to choose the best pipe diameter in a process design. A smaller diameter means a less expensive pipe and lower in-pipe inventory cost, but it means a larger pump. An initial OF choice might be to minimize capital. However, reconsideration from a business investment view might reveal that operating costs associated with pumping power and maintenance are also important issues, and perhaps net present value (NPV) is a right way to combine initial capital with future expenses. Then, reconsideration might bring understanding of the sensitivity of the optimum solution to uncertainty in the "givens," which will lead to a refinement of the OF to represent the 95% worst case of the NPV in a Monte Carlo analysis, making it a stochastic function. Risk might then be perceived as an additional issue, and the OF might be split into a multi-objective version (risk and NPV) that provides a non-dominated set of solutions for a user to select a best for the particular situation. Finally, the user might realize that pipe comes in discrete diameter values and that the pipe diameter is not a continuous-valued number. This application might have evolved from an initial simple deterministic (textbook example) case to a complicated application, classified as mixed integer, stochastic, and multi-objective.

This book will address how to develop the OF and will show examples from a wide range of applications.

Choices

The choices a user has (you may call these inputs, decisions, degrees of freedom, or independent variables) to change things toward the best outcome are termed decision variables (DVs).

In regression DVs are the model coefficient values. In product design DVs could be polymer type, blend concentration, operating a process, color, or shape. In process design DVs could be the pipe diameters and pump sizes. In flying aircraft, the DVs would be the stick, throttle, and pedal positions. In control and scheduling, in operating a business, the DVs would be the future plan for both the timing and magnitude of the actions. Alternately, the DVs might be the coefficients in an equation that would define the future schedule for control actions. Again, there are many possibilities for how to choose the DVs; and the user choices impact efficiency of solution, the appropriate optimizer algorithm, and precision of solution. The book will also address such issues.

Organization

As with most books on engineering optimization, this one describes and develops many common algorithms. It starts with simple univariate (line) search approaches and progresses to multivariable and multiplayer approaches. I do not seek to cover every version, or every method. I use archetypical examples of the many approaches, from which readers can grasp the concepts of other methods. Book topics include gradient based, Newton's, and blends such as Levenberg–Marquardt. They include surrogate function methods to characterize the "surface" such as successive quadratic. They include direct searches such as a simple heuristic cyclic, Hooke–Jeeves, and Nelder–Mead. They include multiplayer mimetic approaches of leapfrogging, particle swarm, and genetic algorithms. They include dynamic programming, in which the DVs are the states, and linear programming that takes advantage of certain structures. The book develops the basic techniques and addresses refinements that improve performance, such as quasi-Newton estimates of the Hessian elements, and grid refinement in dynamic programming.

The book provides a guide to match optimization procedures with features of the application such as discontinuities, flat spots, nearly flat spots, constraints, multiple optima, stochastic responses, parameter correlation, etc. Several sections discuss the issues that certain OF features create. Other sections are devoted to the analysis of the optimizers for precision, accuracy, global identity, work to converge, and robustness. Another section reveals sensitivity to user parameters such as contraction and expansion coefficients, thresholds, triggers, etc. A user needs to understand which optimizer is appropriate for which application and how to make the best choice of optimizer parameter values.

The book also addresses choices of convergence criteria that are appropriate for the application and for the optimizer. For example, in choosing thresholds on the DV as the convergence criteria (which is common practice), the user should use propagation of uncertainty to project the DV tolerance on the OF. As a contrasting example, in optimizing results of either experimental outcomes or a Monte Carlo stochastic simulation, the optimizer needs to stop when the noisy response is not making improvement relative to the noise amplitude.

The book is aimed at engineering applications, where optimization is essential for model development, product design, process and device design, dynamic system control, or system operation. However, the applications of optimization extend into all aspects of our lives from purchasing choices to investment choices, to career planning, and to dressing for a desired impact. The reader should be able to extend the guidance of the book to both personal and other professional decisions.

Rationale for the Book

Optimization is ages old. Prior to calculus, optimization was empirical, guided by heuristics and experience. Improvement was by a direct search, one that only uses the OF value and not the derivative information. The mathematics of calculus, however, created a new era, and Simpson (1740) extended Newton's root finding (1685) to the derivative of the function to find the optimum. Cauchy's sequential line search appeared in 1847. Modest technique progress continued through about 1944, at which time the power of the digital computer led to both practical applications and an explosion in the development of diverse techniques. In 1955 Levenberg blended "Newton's" with incremental steepest descent to spawn many approaches to using both the gradient and Hessian to guide sequential improvements in the trial solution. Advances continued to capitalize on computational power. Then the 1960s gave rise to mimetic multiparticle algorithms and multi-objective applications.

In that brief historical overview, gradient-based techniques replaced the precalculus era direct search techniques. Gradient-based techniques remain the mainstay of texts. However, the power of the digital computer is permitting new direct search techniques such as particle swarm, genetic programming, and leapfrogging to outperform gradient-based techniques on nonlinear and stochastic applications with discontinuities—today's relevant problems. One reason for the book is to promote the use of the new direct search techniques.

Most books on engineering optimization focus on the optimization algorithms. However, most users will not write the code; they will buy it. Of more need for a user is instruction on how to create an appropriate OF, how to choose DVs, how to identify and incorporate constraints, how to define convergence, and how to determine the number of independent starts needed to ensure that the global is found. This book seeks to fill in those application essentials.

I developed and used optimization throughout my initial 13-year career in the industry. However, my college preparation for the engineering career did not teach me what I needed to know about how to create and evaluate optimization applications. I recognized that my fellow engineers, regardless of their *alma mater*, were also underprepared. We had to self-learn what was needed. Recognizing the centrality of optimization to engineering analysis, I have continued to explore its application and technique development in my subsequent 30-year academic career.

This book is based on college and professional training courses that I've offered and is a collection of what I consider to be best practices in engineering optimization. It includes the material I wish I had known when starting my engineering career, and I hope the book is useful for the readers.

Target Audience

The examples and discussion presume basic understanding of engineering models, statistics, calculus, and computer programming. This book will have enough details, explicit equation derivations, and examples to be useful either as an introductory course or for self-study.

The book is aimed at a bachelors, or higher, graduate of engineering or a mathematical science (physics, chemistry, statistics, computer science), who has had an undergraduate course in calculus, mathematical models, statistics, and computer programming. However, upper-level undergraduates have been successful in my course. The reader could be either a student or a practicing engineer or scientist.

Presentation Style

In my experience, students cannot grasp the depth of one topic in isolation of the others. Depth in understanding two-dimensional (2-D) OF surface features is required to be able to relate to N-D issues. An initial understanding of the optimization algorithms is required to be able to set up the application OF and DVs. An understanding of the application is required to be able to choose the appropriate convergence criterion and thresholds. Accordingly, I start the book with elementary versions of each of the aspects of optimization in one-dimensional applications, demonstrate the whole of the applications on several case studies to reveal issues, then return to each item in more depth, and demonstrate the improvements of the second-level techniques in 2-D applications, discuss issues, and then extrapolate to N-D implementations.

I offer the reader with software (and access to computer code through my website www.r3eda.com) to execute key operations. Although there are many strong programming environments, the code is written in Excel VBA (Visual Basic for Applications), which is widely accessible. The book includes a listing of the code for the techniques.

A unique feature of the book is the "takeaway" sections associated with the chapters, which summarize the methods of choice using a practical, applications, utility perspective. This is intended as a user's how-to book grounded in fundamentals, not as a math-analysis-of-the-fundamentals book. However, relevant properties of the optimization problems will be mathematically analyzed, the optimization algorithms will be developed from theory, propagation of uncertainty will be related to choices, and the book contains some proofs related to surface analysis and OF transformations.

Acknowledgments

I am especially grateful to Junyi Li, Balashakar (Soundar) Ramchandran, Ming Su, and Scott Essner, who, as graduate students, brought and revealed to me essential optimization concepts. I am also grateful to many other graduate students who worked with me in exploring and developing understanding of optimization. In reverse chronological order, thanks to Anand Govindarajan, Upasana Manimegalai-Sridhar, Thomas Hays, Haoxian Chen, Chetan Chandak, Prithwijit Ghoshal, Judson Wooters, Gaurav Aurora, Samuel O. Owusu, Venkat Padmanabhan, Solomon Gebreyohannes, Preetica Kumar, Naggappan Muthiah, Nitin Sharma, S. Samir Alam, Qing Li, Jing Ou, Siva Natarajan, Sandeep Chandran, Ganesh Narayanswamy, Mahesh S. Iyer, R. Paul Bray, Abrahao Naim Neto, Ganesh Venkataramanan, Songling Cao, Jin Yong Choi, and Vikram Singh.

I consider myself very fortunate to have been granted the health and ability to enjoy, and now to relay, many experiences and a developing understanding related to optimization. I count my industrial application experience to be as valuable as my academic research investigations. Both are essential for the creation of this book.

Other authors have provided books that have been very valuable to my understanding. I recommend these publications: Ravindran, Ragsdell, and Reklaitis, *Engineering Optimization—Methods and Applications*, Wiley, 2006; Beveridge and Schechter, *Optimization: Theory and Practice*, McGraw-Hill, 1970; Edgar, Himmelblau, and Lasdon, *Optimization of Chemical Processes*, McGraw-Hill, 2001; Snyman, *Practical Mathematical Optimization*, Springer, 2005; Hillier and Lieberman, *Introduction to Operations Research*, McGraw-Hill, 2001; Nocedal and Wright, *Numerical Optimization*, Springer-Verlag, 1999; and Rao, *Engineering Optimization: Theory and Practice*, 4th Edition, Wiley, 2009.

As a professor, funding is essential to enable research, investigation, discovery, and the pursuit of creativity. I am grateful to both the Edward E. & Helen Turner Bartlett Foundation and the Amoco Foundation (now BP) for funding endowments for academic chairs. I have been fortunate to be the chair holder for one or the other, which means that I was permitted to use some proceeds from the endowment to attract and support graduate students who could pursue ideas that did not have traditional research support. This book presents many of the techniques explored, developed, or tested by graduate students. Similarly, I am grateful to the National Science Foundation Industry–University Cooperative Research Centers Program and to a number of industrial sponsors of my graduate program who recognized the importance of applied research and its role in workforce development. These include Amoco, Arco Exploration & Production, Aspen Technologies, Cargill, Chevron Phillips, Diamond Shamrock, Dow Chemical, ExxonMobil, Fina, Gensym, Hoechst Celanese, IMC Agrico, Johnson Yokogawa, LAM, Monsanto, Pavilion Tech, Phillips 66, Tennessee Eastman, Texas Instruments, Union Carbide, and Valero.

Career accomplishments of any one person are the result of the many people who nurtured and developed the person. I am of course grateful to my parents, teachers, and friends, but mostly to Donna, who has encouraged and enabled my work initiatives (really just play and hobbies), as well as appropriately guiding my growth.

Nomenclature

Acronyms

ANOFE	average number of function evaluations
ARIMA	autoregressive integrated moving average
ARMA	autoregressive moving average
CDF	cumulative distribution function
CHD	cyclic heuristic direct
CSLS	Cauchy's sequential line search
D	number of decision variables, the optimization dimension
DCFRR	discounted cash flow rate of return
DE	differential evolution
DMC	Dynamic Matrix Control
DV	decision variable(s), or its value(s)
DV^*	optimum value of the decision variable(s)
EC	equal concern factor (the weights relative importance of additive values)
EHS&LP	Environmental, Health, Safety, and Loss Prevention
FL	fuzzy logic
EPA	Environmental Protection Agency
FOPDT	first-order plus deadtime
GA	genetic algorithm
GRG	generalized reduced gradient
HJ	Hooke–Jeeves
ISD	incremental steepest descent
J	objective function
K.I.S.S.	keep it simple and safe
LF	leapfrogging
LHS	left-hand side
LM	Levenberg–Marquardt
LTROA	long-term return on assets
NM	Nelder–Mead simplex
NN	neural network
NOFE	number of function evaluations
NPV	net present value

NR	Newton–Raphson
$O(x)$	on the order of the value of x
ODE	ordinary differential equation
OF	objective function, or its value
OF*	optimal value of the objective function
OSU	Oklahoma State University
PBT	payback time
PDE	partial differential equation
pdf	probability density function
PNOFE	probable number of function evaluations
PS	particle swarm
PSO	particle swarm optimization
RHS	right-hand side
rms	root-mean-square value
SOPDT	second-order plus deadtime
ST	subject to, also S.T.
SQ	successive quadratic
SS	steady state
SSD	sum of squared deviations, alternately just sum of squares
TS	trial solution, a set of DV values
TS	transient state
TSP	traveling salesman problem
w.r.t.	with respect to

Definitions

a posteriori	A Latin term, indicating “after it has been done.” A choice made after event outcomes have been observed.
a priori	A Latin term, indicating “before doing it.” A choice made before the event, from earlier experience or understanding, not after observation.
Bottom line	The reveal of comprehensive issues.
CDF(OF)*	The cumulative distribution function is a useful tool for visualizing the probability of finding the global optimum and the certainty of its location.
Constraints	These are what should not, or must not, be violated. Some constraints must not be violated because the violation may be catastrophic, like an implosion. Alternately, if other constraints are violated, there might just be a modest penalty. Constraints can be inequality or equality. If equality relations in the DVs, they can be used to reduce the number of DVs if the structure permits solving for one DV given the others. “Hard” constraints are of the “must not” violate category and limit DV TS choices. “Soft” means that the objective function is given a penalty for the

constraint violation, which softens the base of the cliff with a curve, which converts the surface to one that is analytically tractable.

Convergence and stopping criteria Something has to indicate that either the optimum is found or the optimizer is hopelessly lost and needs to be stopped. *Convergence* means that the DV* values have been found, that they will be effectively unchanged in sequential trial solutions, and that the optimizer can be stopped, claiming convergence or close enough proximity to the ideal DV*. The tolerance, or precision, or accuracy could be specified on the change in the DVs or the change in the OF value, or on any of many more complex relations such as the maximum impact on the OF value due to range of the DVs, or when the improvement in the OF is inconsequential to the uncertainty on the OF. The optimizer iterations are stopped when convergence is claimed; but if it has been running excessively without convergence, it could be stopped and "no convergence" reported. *Stopping* criteria could be on the number of iterations, run time, or such indicators that more computation will be futile. Both the criteria and the threshold values for both stopping and convergence action are user choices and are critical to the validity of the reported solution.

Convex The feasible space boundary is always open to, curved to enclose, the feasible DV space. Take any two points in feasible DV space and draw a straight line between them. If every point on the line is within the feasible space, for any pair of feasible points, then the OF is convex. For instance, a rectangle, circle, and ellipse enclose feasible spaces that are convex. However, a kidney bean or a boomerang shape is not. And a circular infeasibility region within a square makes the application non-convex.

Customer A person, or entity, or group that has a legitimate claim related to desirability or undesirability of your application. Alternately termed a stakeholder.

Derivative evaluation Gradient-based optimizers require the values of the derivatives and/or second derivatives. If the OF is a relatively simple function, then the derivative formulas can be analytically derived, and values explicitly calculated. But in meaningful applications the OF is usually a procedure, in which case derivatives need to be estimated numerically. Should one use the central difference? Or should one use a forward or backward approximation? The central difference requires an additional OF evaluation but provides a better estimate of the true derivative. But is the work worth the benefit? A forward Δx_i difference might cross over a constraint into an infeasible region. What then? What should be the Δx_i value? Too small and it will cause truncation error. Too large and the numerical procedure will misrepresent the local surface.

DV (decision variables) These are what you can change to improve the OF value. There may be several or just one. The DVs might be independent or interrelated (reflux must be less than vapor boilup—constrained by each other). They may have rate-of-change constraints. The DVs might be scheduled with another variable, as in the Goddard problem of choosing rocket thrust to maximize height (thrust could be scheduled with time, height, or remaining fuel). The DVs might be coefficients in an equation that relates the process inputs to the state variable that they are scheduled onto. The DVs might be a continuum variable, an integer or discretized variable, or a class or category. The DV choice is critical. It must match the customers' perception of the flexibility that you have within the application context. Further, your selection of number of stages in scheduling or functional relations in converting DVs to equation coefficients has a substantial impact on both the DV^* and OF^* values and computational work.

Evaluation of optimizers — We want a high probability that a procedure will find the global optima. And we want to find OF^* with fewest number of function evaluations, greatest robustness to surface aberrations and least user involvement and dependency on user choices. I'll use number of function evaluations (NOFE) as a measure of work and combine this with probability of finding the global optima as an indicator of success to get the probable NOFE (PNOFE) as a primary evaluator of optimizer performance.

Feasible — A DV value that neither violates a hard constraint nor leads to a computer execution error (overflow, divide by zero, log of a negative, subscript out of range, etc.).

Givens — These diverse aspects are the basis, assumptions, conditions, procedures, models, etc. in the analysis. Consider these givens in a problem statement, "The glass is half full of water, how long does it take to evaporate?" Is the temperature exactly known and constant over the evaporation period? Is the relative humidity exactly known? How can a glass be exactly half filled? Are there any air currents over the glass? What is the impurity content (salts, dissolved CO_2) of the water? What model is exactly right for the vapor–liquid equilibrium, equation of state, or meniscus effect calculations? There is uncertainty in all of these givens. The givens are not truths. They are just approximations. The uncertainty in the givens has an impact on the application solution, and the consequential uncertainty in the DV^* and OF^* should be acknowledged.

Greedy algorithm — Take the local best action. Only look at the current situation, not the future implications of the action. For example, your car may be in the garage, but you cannot walk directly toward it because of the wall. You need to take a longer constraint free path through the door to the garage, which also includes the stage of opening the door. In the traveling salesman problem, a greedy algorithm is the heuristic of going to the next closest city.

Initialization — This refers to both the initial trial solution and the optimizer parameter values. If you start in St. Louis, the downhill path takes you to New Orleans at sea level. But if you start west of Las Vegas, downhill moves you to Death Valley at an elevation below sea level. In a multi-optima surface, the optimizer solution depends on where you start. The exact path an optimizer travels also depends on coefficients in the optimizer that control step size and on other user choices such as whether to use a central difference or backward difference numerical estimate of the derivative. Such coefficients will affect both the number of function evaluations to reach the optimum and the TS value when convergence is claimed.

Number of independent trials If local optima, constraints, ridges, etc. can trap a trial solution, then run multiple optimizer trials from independent DV initializations. The more independent trials you run, the greater is the probability that you'll find the global.

Maximum or minimum Since MAX J = OF provides the same DV* as MIN J = –OF, it is irrelevant whether the optimizer is set to minimize (find lower values of the negative OF) or maximize (find higher OF values). Following convention, without loss of generality, this book seeks to minimize.

Model There must be mechanisms for determining the OF value and constraint conditions from the DV TS values. The models have many choices (ideal gas law or virial equation of state, Hooke's law or a viscoelastic relation). And the models need to include how goodness is evaluated, which means the models reflect human values as well as engineering analysis. The mechanism might include experimental testing, followed by constructing assessment metrics.

Multiple objectives Many applications seek a balance of opposing ideals (e.g., profit vs. waste, personal image vs. cost, perfection of fit vs. number of coefficients). There are many diverse metrics of application desirability. Most optimizers need a single OF value and seek to optimize it. In such cases the multiple objectives need to be combined into a single function. Usually the separate objective functions are added and weighted to unify dimensions and scale to an appropriate balance. For this approach, I like the equal concern approach for generating weighting factors. But sometimes the diverse OF criteria are multiplied. Alternately, the OF functions are kept separate, and in a Pareto analysis, all the non-dominated TSs are kept as the Pareto optimal set of possibilities for a user to select which is appropriate for a particular situation.

Non-convex The feasible space boundary is not always open to, curved to enclose, the feasible DV space. Take any two points in feasible DV space and draw a straight line between them. If any point on the line is outside of the feasible space, for any pair of feasible points, then the OF is non-convex. For instance, a rectangle, circle, and ellipse enclose feasible spaces that are convex. However, a kidney bean or a boomerang shape is not. And a circular infeasibility region within a square makes the application non-convex.

OF (objective function) This is the procedure for determining the measure of goodness, the assessment of desirability. Defining it is a critical step, because the metrics you use to assess value determines the solution you get. You need to be sure that you are defining the right metrics of goodness. For instance, in an economic optimization for a process design, with pipe diameter as the DV, you get different answers if the objective is to minimize capital, expenses, net present value, or probable risk. You also get different answers if the business model is for a 5-year period or for a 20-year period. As another example, in placing surveillance devices, you get different answers if the objective is to maximize observation of the boundary or to maximize the minimum coverage of the internal area. Also, often there are competing issues that need to be balanced with appropriate weighting. In all, it is essential to use metrics, calculation models, and weighting that reflect the customers' values and context for use.

OF transformations — For an OF with positive values, the minimum of the log of an OF has the same DV^* values as the minimum of the original OF. DV^* is unchanged with an OF that is transformed with a positive definite function. Also, minimizing the OF has the same DV^* as maximizing the negative of the OF. OF transformations can be conveniences in scaling for display, converting max to min, or simplifying of the concepts. They could convert an OF to a functional form that is more quadratic-like, which may improve speed of convergence for Newton-like algorithms, but they have no impact on speed in direct search algorithms. It does not affect the presence of local optima. Transformations might distort OF-based convergence criteria. As a caution, functional transforming individual elements in the OF or changing the weighting of individual terms will change the DV^* value.

OF value (objective function value) — This is the value that your procedure returns.

Optimizer — This is the procedure for iteratively improving the TS toward DV^*. If an analytical method is possible, the solution directly calculates DV^*. However, usually, practical applications are analytically intractable, and an iterative approach is needed. There are two general approaches. Gradient-based approaches use the local surface slope to indicate the direction of steepest descent and move TS in that direction. Incremental steepest descent and Cauchy's are of this type. Newton-type approaches use the surface slope and also the second derivatives to adjust the leap-to TS. They are derived by a truncation of a Taylor series model of the surface to quadratic terms. Levenberg–Marquardt combines both incremental steepest descent and Newton's. Successive quadratic generates a local quadratic surrogate model of the surface and then leaps to the DV^* of the surrogate model. Although SQ explicitly uses neither gradient nor Hessian elements, it is equivalent to Newton's in performance. Both are based on a local quadratic model of the surface. Surface aberrations of flat spots or discontinuities are a problem for those approaches that are based on a rational model of a continuous and quadratic surface. Contrasting the gradient-based approaches are the direct search approaches that do not use suppositions of the surface structure. Direct searches only consider the OF value and constraint "PASS" or "FAIL" status and move the TS using heuristic or stochastic rules. Cyclic Heuristic, Hooke–Jeeves, Nelder–Mead simplex, particle swarm, genetic algorithms, differential evolution, and leapfrogging are direct search algorithms. Because these do not leap-to the surrogate model optimum, but approach it gradually, one would think that the gradient-based techniques are faster. However, to numerically evaluate the local surface in a 3-DV problem, SQ and Levenberg–Marquardt each need 10 local OF evaluations; the direct search approaches use many fewer and in many cases converge with fewer NOFE. Further, the direct searches are more robust to surface features. Some older direct search algorithms (HJ, NM) are a single trial solution search (one initial guess moves locally downhill). However, the more recent ones (PSO, GA, DE, LF) use multiple "simultaneous" trial solutions, and the collective evidence of the surface provides a "global" view of where all should go.

Risk — A penalty for a possible undesired event. Risk is the probability of the undesired event times the cost or equivalent impact of the event.

Stakeholder — A person, or entity, or group that has a legitimate claim related to desirability or undesirability of your application. Alternately termed a customer.

Surface aberrations — It would be nice if the optimization application created a smooth TS path to the bottom of the valley, so that gradient-based approaches could guide the DV TS to the optimum. But applications have a variety of surface aberrations that confound a search. It is easiest to visualize surfaces as a 3-D response to 2 DVs and to use familiar terms to describe surface features; but the concepts and issues are scalable to *N*-D applications. Flat sections in the surface arise because of discretization in DV size or category or OF rank or category. On a flat section the optimizer cannot determine what direction is down. Cliffs and slope discontinuities arise with constraints or with switches in units or modes (turbulent to laminar, three to four parallel units), which may be expressed as IF-THEN conditionals in the model. On either side of a steep valley, the slope on the walls points across the valley, and the optimizer will zigzag across the valley, not move along the ridge. Multiple optima mean that there are multiple local minima; and at any one, all directions seem worse. These often arise in nonlinear models, and you don't want an optimizer to get stuck in a local optimum and report that value without acknowledging the other locations. The derivative is zero at saddle points, maximum, and minimum, and some gradient-based optimizers seek any such point, not just the optima. Stochastic surfaces are the result of experimental or Monte Carlo simulation responses. In a stochastic surface, replicate DV trials do not return the exact same OF value. If a new OF value is better, does that mean that the TS moved in the right direction or simply that the vagaries of noise made it seem better?

Surrogate functions	These are useful for optimizer testing. They are simple to code, quick to compute, and reveal the issues that serious applications would present.
TS (trial solution)	A set of values for the DVs. A "guess" at the DV* values. Optimizers progressively, iteratively, seek TS values that lead to a better OF.
Uncertainty	There are always "givens" in the application. Givens are the specifications that you are given by the problem statement. Consider this statement: "The weight of the truck is 1.23 tons and the tires are inflated to 55 psig. What is the optimum speed?" Will the tire pressure always be exactly 55 psig? Will other conditions be known with certainty? What about road elevation, barometric pressure, air relative humidity, fuel BTU content, wind, temperature, and so on? The DV* value may be exactly right for a particular set of givens, but might not be the overall best for the range of possible conditions and influences. One should investigate the impact of uncertainty of the "givens" on the DV* value.

Symbols

$a, b, c, d, \ldots$	model coefficients, scalar variables
$w, x, y, z, \ldots$	model variables, scalar variables
$\tilde{y}$	modeled response value (as opposed to the measured value)
$\boldsymbol{x}$	a vector of x-elements
$\mathbf{M}$	a matrix of m elements
$\mathbf{H}$	the Hessian matrix of second-order partial derivatives
$\forall$	for all
Δ	delta, a small increment
∇	gradient, or Jacobean, the $\partial/\partial x$ operator
:=	the assignment statement in a computer program, to differentiate from an algebraic =
$\lvert x \rvert$	absolute value of a scalar variable
$\lVert \boldsymbol{x} \rVert$	magnitude of a vector $\sqrt{\sum x_i^2}$
$\parallel$	parallel
$\perp$	perpendicular
!	factorial
$O(3)$	on the order of three

About the Companion Website

This book is accompanied by a companion website:

www.wiley.com/go/rhinehart/engineeringoptimization

The website includes:

- Techniques
- Excel VBA programs
- Software