

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



This edition first published 2018

© 2018 R. Russell Rhinehart

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

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, except as permitted by law. Advice on how to obtain permission to reuse material from this title is available at <http://www.wiley.com/go/permissions>.

The right of Robert Russell Rhinehart, II to be identified as the author of this work has been asserted in accordance with law.

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.

Limit of Liability/Disclaimer of Warranty

While the publisher and authors have used their best efforts in preparing this work, they make no representations or warranties with respect to the accuracy or completeness of the contents of this work and specifically disclaim all warranties, including without limitation any implied warranties of merchantability or fitness for a particular purpose. No warranty may be created or extended by sales representatives, written sales materials or promotional statements for this work. The fact that an organization, website, or product is referred to in this work as a citation and/or potential source of further information does not mean that the publisher and authors endorse the information or services the organization, website, or product may provide or recommendations it may make. This work is sold with the understanding that the publisher is not engaged in rendering professional services. The advice and strategies contained herein may not be suitable for your situation. You should consult with a specialist where appropriate. Further, readers should be aware that websites listed in this work may have changed or disappeared between when this work was written and when it is read. Neither the publisher nor authors shall be liable for any loss of profit or any other commercial damages, including but not limited to special, incidental, consequential, or other damages.

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 | <i>xix</i> |
| Acknowledgments | <i>xxvii</i> |
| Nomenclature | <i>xxix</i> |
| About the Companion Website | <i>xxxvii</i> |

Section 1 Introductory Concepts 1

| | | |
|----------|----------------------------------------------------------|-----------|
| 1 | Optimization: Introduction and Concepts | 3 |
| 1.1 | Optimization and Terminology | 3 |
| 1.2 | Optimization Concepts and Definitions | 4 |
| 1.3 | Examples | 6 |
| 1.4 | Terminology Continued | 10 |
| 1.4.1 | Constraint | 10 |
| 1.4.2 | Feasible Solutions | 10 |
| 1.4.3 | Minimize or Maximize | 11 |
| 1.4.4 | Canonical Form of the Optimization Statement | 11 |
| 1.5 | Optimization Procedure | 12 |
| 1.6 | Issues That Shape Optimization Procedures | 16 |
| 1.7 | Opposing Trends | 17 |
| 1.8 | Uncertainty | 20 |
| 1.9 | Over- and Under-specification in Linear Equations | 21 |
| 1.10 | Over- and Under-specification in Optimization | 22 |
| 1.11 | Test Functions | 23 |
| 1.12 | Significant Dates in Optimization | 23 |
| 1.13 | Iterative Procedures | 26 |
| 1.14 | Takeaway | 27 |
| 1.15 | Exercises | 27 |
| 2 | Optimization Application Diversity and Complexity | 33 |
| 2.1 | Optimization | 33 |
| 2.2 | Nonlinearity | 33 |
| 2.3 | Min, Max, Min–Max, Max–Min, ... | 34 |
| 2.4 | Integers and Other Discretization | 35 |

| | | |
|-----------------------------------------------|------------------------------------------------------------------------|-----------|
| 2.5 | Conditionals and Discontinuities: Cliffs Ridges/Valleys | 36 |
| 2.6 | Procedures, Not Equations | 37 |
| 2.7 | Static and Dynamic Models | 38 |
| 2.8 | Path Integrals | 38 |
| 2.9 | Economic Optimization and Other Nonadditive Cost Functions | 38 |
| 2.10 | Reliability | 39 |
| 2.11 | Regression | 40 |
| 2.12 | Deterministic and Stochastic | 42 |
| 2.13 | Experimental w.r.t. Modeled OF | 43 |
| 2.14 | Single and Multiple Optima | 44 |
| 2.15 | Saddle Points | 45 |
| 2.16 | Inflections | 46 |
| 2.17 | Continuum and Discontinuous DVs | 47 |
| 2.18 | Continuum and Discontinuous Models | 47 |
| 2.19 | Constraints and Penalty Functions | 48 |
| 2.20 | Ranks and Categorization: Discontinuous OFs | 50 |
| 2.21 | Underspecified OFs | 51 |
| 2.22 | Takeaway | 51 |
| 2.23 | Exercises | 51 |
| 3 | Validation: Knowing That the Answer Is Right | 53 |
| 3.1 | Introduction | 53 |
| 3.2 | Validation | 53 |
| 3.3 | Advice on Becoming Proficient | 55 |
| 3.4 | Takeaway | 56 |
| 3.5 | Exercises | 57 |
| | | |
| Section 2 Univariate Search Techniques | | 59 |
| | | |
| 4 | Univariate (Single DV) Search Techniques | 61 |
| 4.1 | Univariate (Single DV) | 61 |
| 4.2 | Analytical Method of Optimization | 62 |
| 4.2.1 | Issues with the Analytical Approach | 63 |
| 4.3 | Numerical Iterative Procedures | 64 |
| 4.3.1 | Newton's Methods | 64 |
| 4.3.2 | Successive Quadratic (A Surrogate Model or Approximating Model Method) | 68 |
| 4.4 | Direct Search Approaches | 70 |
| 4.4.1 | Bisection Method | 70 |
| 4.4.2 | Golden Section Method | 72 |
| 4.4.3 | Perspective at This Point | 74 |
| 4.4.4 | Heuristic Direct Search | 74 |
| 4.4.5 | Leapfrogging | 76 |
| 4.4.6 | LF for Stochastic Functions | 79 |
| 4.5 | Perspectives on Univariate Search Methods | 82 |

| | | |
|----------|------------------------------------------------------------|------------|
| 4.6 | Evaluating Optimizers | 85 |
| 4.7 | Summary of Techniques | 85 |
| 4.7.1 | Analytical Method | 86 |
| 4.7.2 | Newton's (and Variants Like Secant) | 86 |
| 4.7.3 | Successive Quadratic | 86 |
| 4.7.4 | Golden Section Method | 86 |
| 4.7.5 | Heuristic Direct | 87 |
| 4.7.6 | Leapfrogging | 87 |
| 4.8 | Takeaway | 87 |
| 4.9 | Exercises | 88 |
| 5 | Path Analysis | 93 |
| 5.1 | Introduction | 93 |
| 5.2 | Path Examples | 93 |
| 5.3 | Perspective About Variables | 96 |
| 5.4 | Path Distance Integral | 97 |
| 5.5 | Accumulation along a Path | 99 |
| 5.6 | Slope along a Path | 101 |
| 5.7 | Parametric Path Notation | 103 |
| 5.8 | Takeaway | 104 |
| 5.9 | Exercises | 104 |
| 6 | Stopping and Convergence Criteria: 1-D Applications | 107 |
| 6.1 | Stopping versus Convergence Criteria | 107 |
| 6.2 | Determining Convergence | 107 |
| 6.2.1 | Threshold on the OF | 108 |
| 6.2.2 | Threshold on the Change in the OF | 108 |
| 6.2.3 | Threshold on the Change in the DV | 108 |
| 6.2.4 | Threshold on the Relative Change in the DV | 109 |
| 6.2.5 | Threshold on the Relative Change in the OF | 109 |
| 6.2.6 | Threshold on the Impact of the DV on the OF | 109 |
| 6.2.7 | Convergence Based on Uncertainty Caused by the Givens | 109 |
| 6.2.8 | Multiplayer Range | 110 |
| 6.2.9 | Steady-State Convergence | 110 |
| 6.3 | Combinations of Convergence Criteria | 111 |
| 6.4 | Choosing Convergence Threshold Values | 112 |
| 6.5 | Precision | 112 |
| 6.6 | Other Convergence Criteria | 113 |
| 6.7 | Stopping Criteria to End a Futile Search | 113 |
| 6.7.1 | N Iteration Threshold | 114 |
| 6.7.2 | Execution Error | 114 |
| 6.7.3 | Constraint Violation | 114 |
| 6.8 | Choices! | 114 |
| 6.9 | Takeaway | 114 |
| 6.10 | Exercises | 115 |

| | | | |
|----------|-----------------------------------------------------------------|---------------------------------------|------------|
| | Section 3 | Multivariate Search Techniques | <i>117</i> |
| 7 | Multidimension Application Introduction and the Gradient | | <i>119</i> |
| 7.1 | Introduction | | <i>119</i> |
| 7.2 | Illustration of Surface and Terms | | <i>122</i> |
| 7.3 | Some Surface Analysis | | <i>123</i> |
| 7.4 | Parametric Notation | | <i>128</i> |
| 7.5 | Extension to Higher Dimension | | <i>130</i> |
| 7.6 | Takeaway | | <i>131</i> |
| 7.7 | Exercises | | <i>131</i> |
| 8 | Elementary Gradient-Based Optimizers: CSLS and ISD | | <i>135</i> |
| 8.1 | Introduction | | <i>135</i> |
| 8.2 | Cauchy's Sequential Line Search | | <i>135</i> |
| 8.2.1 | CSLS with Successive Quadratic | | <i>137</i> |
| 8.2.2 | CSLS with Newton/Secant | | <i>138</i> |
| 8.2.3 | CSLS with Golden Section | | <i>138</i> |
| 8.2.4 | CSLS with Leapfrogging | | <i>138</i> |
| 8.2.5 | CSLS with Heuristic Direct Search | | <i>139</i> |
| 8.2.6 | CSLS Commentary | | <i>139</i> |
| 8.2.7 | CSLS Pseudocode | | <i>140</i> |
| 8.2.8 | VBA Code for a 2-DV Application | | <i>141</i> |
| 8.3 | Incremental Steepest Descent | | <i>144</i> |
| 8.3.1 | Pseudocode for the ISD Method | | <i>144</i> |
| 8.3.2 | Enhanced ISD | | <i>145</i> |
| 8.3.3 | ISD Code | | <i>148</i> |
| 8.4 | Takeaway | | <i>149</i> |
| 8.5 | Exercises | | <i>149</i> |
| 9 | Second-Order Model-Based Optimizers: SQ and NR | | <i>155</i> |
| 9.1 | Introduction | | <i>155</i> |
| 9.2 | Successive Quadratic | | <i>155</i> |
| 9.2.1 | Multivariable SQ | | <i>156</i> |
| 9.2.2 | SQ Pseudocode | | <i>159</i> |
| 9.3 | Newton–Raphson | | <i>159</i> |
| 9.3.1 | NR Pseudocode | | <i>162</i> |
| 9.3.2 | Attenuate NR | | <i>163</i> |
| 9.3.3 | Quasi-Newton | | <i>166</i> |
| 9.4 | Perspective on CSLS, ISD, SQ, and NR | | <i>168</i> |
| 9.5 | Choosing Step Size for Numerical Estimate of Derivatives | | <i>169</i> |
| 9.6 | Takeaway | | <i>170</i> |
| 9.7 | Exercises | | <i>170</i> |

| | | |
|-----------|---------------------------------------------------------------------------|------------|
| 10 | Gradient-Based Optimizer Solutions: LM, RLM, CG, BFGS, RG, and GRG | 173 |
| 10.1 | Introduction | 173 |
| 10.2 | Levenberg–Marquardt (LM) | 173 |
| 10.2.1 | LM VBA Code for a 2-DV Case | 175 |
| 10.2.2 | Modified LM (RLM) | 176 |
| 10.2.3 | RLM Pseudocode | 177 |
| 10.2.4 | RLM VBA Code for a 2-DV Case | 178 |
| 10.3 | Scaled Variables | 180 |
| 10.4 | Conjugate Gradient (CG) | 182 |
| 10.5 | Broyden–Fletcher–Goldfarb–Shanno (BFGS) | 183 |
| 10.6 | Generalized Reduced Gradient (GRG) | 184 |
| 10.7 | Takeaway | 186 |
| 10.8 | Exercises | 186 |
| | | |
| 11 | Direct Search Techniques | 187 |
| 11.1 | Introduction | 187 |
| 11.2 | Cyclic Heuristic Direct (CHD) Search | 188 |
| 11.2.1 | CHD Pseudocode | 188 |
| 11.2.2 | CHD VBA Code | 189 |
| 11.3 | Hooke–Jeeves (HJ) | 192 |
| 11.3.1 | HJ Code in VBA | 195 |
| 11.4 | Compare and Contrast CHD and HJ Features: A Summary | 197 |
| 11.5 | Nelder–Mead (NM) Simplex: Spendley, Hext, and Himsworth | 199 |
| 11.6 | Multiplayer Direct Search Algorithms | 200 |
| 11.7 | Leapfrogging | 201 |
| 11.7.1 | Convergence Criteria | 208 |
| 11.7.2 | Stochastic Surfaces | 209 |
| 11.7.3 | Summary | 209 |
| 11.8 | Particle Swarm Optimization | 209 |
| 11.8.1 | Individual Particle Behavior | 210 |
| 11.8.2 | Particle Swarm | 213 |
| 11.8.3 | PSO Equation Analysis | 215 |
| 11.9 | Complex Method (CM) | 216 |
| 11.10 | A Brief Comparison | 217 |
| 11.11 | Takeaway | 218 |
| 11.12 | Exercises | 219 |
| | | |
| 12 | Linear Programming | 223 |
| 12.1 | Introduction | 223 |
| 12.2 | Visual Representation and Concepts | 225 |
| 12.3 | Basic LP Procedure | 228 |
| 12.4 | Canonical LP Statement | 228 |

| | | |
|-----------|--------------------------------------------------------|------------|
| 12.5 | LP Algorithm | 229 |
| 12.6 | Simplex Tableau | 230 |
| 12.7 | Takeaway | 231 |
| 12.8 | Exercises | 231 |
| 13 | Dynamic Programming | 233 |
| 13.1 | Introduction | 233 |
| 13.2 | Conditions | 236 |
| 13.3 | DP Concept | 237 |
| 13.4 | Some Calculation Tips | 240 |
| 13.5 | Takeaway | 241 |
| 13.6 | Exercises | 241 |
| 14 | Genetic Algorithms and Evolutionary Computation | 243 |
| 14.1 | Introduction | 243 |
| 14.2 | GA Procedures | 243 |
| 14.3 | Fitness of Selection | 245 |
| 14.4 | Takeaway | 250 |
| 14.5 | Exercises | 250 |
| 15 | Intuitive Optimization | 253 |
| 15.1 | Introduction | 253 |
| 15.2 | Levels | 254 |
| 15.3 | Takeaway | 254 |
| 15.4 | Exercises | 254 |
| 16 | Surface Analysis II | 257 |
| 16.1 | Introduction | 257 |
| 16.2 | Maximize Is Equivalent to Minimize the Negative | 257 |
| 16.3 | Scaling by a Positive Number Does Not Change DV* | 258 |
| 16.4 | Scaled and Translated OFs Do Not Change DV* | 258 |
| 16.5 | Monotonic Function Transformation Does Not Change DV* | 258 |
| 16.6 | Impact on Search Path or NOFE | 261 |
| 16.7 | Inequality Constraints | 263 |
| 16.8 | Transforming DVs | 263 |
| 16.9 | Takeaway | 263 |
| 16.10 | Exercises | 263 |
| 17 | Convergence Criteria 2: N-D Applications | 265 |
| 17.1 | Introduction | 265 |
| 17.2 | Defining an Iteration | 265 |
| 17.3 | Criteria for Single TS Deterministic Procedures | 266 |
| 17.4 | Criteria for Multiplayer Deterministic Procedures | 267 |
| 17.5 | Stochastic Applications | 268 |
| 17.6 | Miscellaneous Observations | 268 |

| | | |
|-------------------------------------------------------------|-------------------------------------------------------------------|------------|
| 17.7 | Takeaway | 269 |
| 17.8 | Exercises | 269 |
| 18 | Enhancements to Optimizers | 271 |
| 18.1 | Introduction | 271 |
| 18.2 | Criteria for Replicate Trials | 271 |
| 18.3 | Quasi-Newton | 274 |
| 18.4 | Coarse–Fine Sequence | 275 |
| 18.5 | Number of Players | 275 |
| 18.6 | Search Range Adjustment | 276 |
| 18.7 | Adjustment of Optimizer Coefficient Values or Options in Process | 276 |
| 18.8 | Initialization Range | 277 |
| 18.9 | OF and DV Transformations | 277 |
| 18.10 | Takeaway | 278 |
| 18.11 | Exercises | 278 |
| | | |
| Section 4 Developing Your Application Statements 279 | | |
| | | |
| 19 | Scaled Variables and Dimensional Consistency | 281 |
| 19.1 | Introduction | 281 |
| 19.2 | A Scaled Variable Approach | 283 |
| 19.3 | Sampling of Issues with Primitive Variables | 283 |
| 19.4 | Linear Scaling Options | 285 |
| 19.5 | Nonlinear Scaling | 286 |
| 19.6 | Takeaway | 287 |
| 19.7 | Exercises | 287 |
| | | |
| 20 | Economic Optimization | 289 |
| 20.1 | Introduction | 289 |
| 20.2 | Annual Cash Flow | 290 |
| 20.3 | Including Risk as an Annual Expense | 291 |
| 20.4 | Capital | 293 |
| 20.5 | Combining Capital and Nominal Annual Cash Flow | 293 |
| 20.6 | Combining Time Value and Schedule of Capital and Annual Cash Flow | 296 |
| 20.7 | Present Value | 297 |
| 20.8 | Including Uncertainty | 298 |
| 20.8.1 | Uncertainty Models | 301 |
| 20.8.2 | Methods to Include Uncertainty in an Optimization | 303 |
| 20.9 | Takeaway | 304 |
| 20.10 | Exercises | 304 |
| | | |
| 21 | Multiple OF and Constraint Applications | 305 |
| 21.1 | Introduction | 305 |
| 21.2 | Solution 1: Additive Combinations of the Functions | 306 |

| | | |
|-----------|-------------------------------------------------------|------------|
| 21.2.1 | Solution 1a: Classic Weighting Factors | 307 |
| 21.2.2 | Solution 1b: Equal Concern Weighting | 307 |
| 21.2.3 | Solution 1c: Nonlinear Weighting | 309 |
| 21.3 | Solution 2: Nonadditive OF Combinations | 311 |
| 21.4 | Solution 3: Pareto Optimal | 311 |
| 21.5 | Takeaway | 316 |
| 21.6 | Exercises | 316 |
| 22 | Constraints | 319 |
| 22.1 | Introduction | 319 |
| 22.2 | Equality Constraints | 320 |
| 22.2.1 | Explicit Equality Constraints | 320 |
| 22.2.2 | Implicit Equality Constraints | 321 |
| 22.3 | Inequality Constraints | 321 |
| 22.3.1 | Penalty Function: Discontinuous | 323 |
| 22.3.2 | Penalty Function: Soft Constraint | 323 |
| 22.3.3 | Inequality Constraints: Slack and Surplus Variables | 325 |
| 22.4 | Constraints: Pass/Fail Categories | 329 |
| 22.5 | Hard Constraints Can Block Progress | 330 |
| 22.6 | Advice | 331 |
| 22.7 | Constraint-Equivalent Features | 332 |
| 22.8 | Takeaway | 332 |
| 22.9 | Exercises | 332 |
| 23 | Multiple Optima | 335 |
| 23.1 | Introduction | 335 |
| 23.2 | Solution: Multiple Starts | 337 |
| 23.2.1 | <i>A Priori</i> Method | 340 |
| 23.2.2 | <i>A Posteriori</i> Method | 342 |
| 23.2.3 | Snyman and Fatti Criterion <i>A Posteriori</i> Method | 345 |
| 23.3 | Other Options | 348 |
| 23.4 | Takeaway | 349 |
| 23.5 | Exercises | 350 |
| 24 | Stochastic Objective Functions | 353 |
| 24.1 | Introduction | 353 |
| 24.2 | Method Summary for Optimizing Stochastic Functions | 356 |
| 24.2.1 | Step 1: Replicate the Apparent Best Player | 356 |
| 24.2.2 | Step 2: Steady-State Detection | 357 |
| 24.3 | What Value to Report? | 358 |
| 24.4 | Application Examples | 359 |
| 24.4.1 | GMC Control of Hot and Cold Mixing | 359 |
| 24.4.2 | MBC of Hot and Cold Mixing | 359 |
| 24.4.3 | Batch Reaction Management | 359 |
| 24.4.4 | Reservoir and Stochastic Boot Print | 361 |
| 24.4.5 | Optimization Results | 362 |

| | | |
|-----------|---------------------------------------------------------------------------|------------|
| 24.5 | Takeaway | 365 |
| 24.6 | Exercises | 365 |
| 25 | Effects of Uncertainty | 367 |
| 25.1 | Introduction | 367 |
| 25.2 | Sources of Error and Uncertainty | 368 |
| 25.3 | Significant Digits | 370 |
| 25.4 | Estimating Uncertainty on Values | 371 |
| 25.5 | Propagating Uncertainty on DV Values | 372 |
| 25.5.1 | Analytical Method | 373 |
| 25.5.2 | Numerical Method | 375 |
| 25.6 | Implicit Relations | 378 |
| 25.7 | Estimating Uncertainty in DV* and OF* | 378 |
| 25.8 | Takeaway | 379 |
| 25.9 | Exercises | 379 |
| 26 | Optimization of Probable Outcomes and Distribution Characteristics | 381 |
| 26.1 | Introduction | 381 |
| 26.2 | The Concept of Modeling Uncertainty | 385 |
| 26.3 | Stochastic Approach | 387 |
| 26.4 | Takeaway | 389 |
| 26.5 | Exercises | 389 |
| 27 | Discrete and Integer Variables | 391 |
| 27.1 | Introduction | 391 |
| 27.2 | Optimization Solutions | 394 |
| 27.2.1 | Exhaustive Search | 394 |
| 27.2.2 | Branch and Bound | 394 |
| 27.2.3 | Cyclic Heuristic | 394 |
| 27.2.4 | Leapfrogging or Other Multiplayer Search | 395 |
| 27.3 | Convergence | 395 |
| 27.4 | Takeaway | 395 |
| 27.5 | Exercises | 395 |
| 28 | Class Variables | 397 |
| 28.1 | Introduction | 397 |
| 28.2 | The Random Keys Method: Sequence | 398 |
| 28.3 | The Random Keys Method: Dichotomous Variables | 400 |
| 28.4 | Comments | 401 |
| 28.5 | Takeaway | 401 |
| 28.6 | Exercises | 401 |
| 29 | Regression | 403 |
| 29.1 | Introduction | 403 |
| 29.2 | Perspective | 404 |
| 29.3 | Least Squares Regression: Traditional View on Linear Model Parameters | 404 |

- 29.4 Models Nonlinear in DV 405
- 29.4.1 Models with a Delay 407
- 29.5 Maximum Likelihood 408
- 29.5.1 Akaho's Method 411
- 29.6 Convergence Criterion 416
- 29.7 Model Order or Complexity 421
- 29.8 Bootstrapping to Reveal Model Uncertainty 425
- 29.8.1 Interpretation of Bootstrapping Analysis 428
- 29.8.2 Appropriating Bootstrapping 430
- 29.9 Perspective 431
- 29.10 Takeaway 431
- 29.11 Exercises 432

Section 5 Perspective on Many Topics 441

- 30 Perspective 443**
- 30.1 Introduction 443
- 30.2 Classifications 443
- 30.3 Elements Associated with Optimization 445
- 30.4 Root Finding Is Not Optimization 446
- 30.5 Desired Engineering Attributes 446
- 30.6 Overview of Optimizers and Attributes 447
- 30.6.1 Gradient Based: Cauchy Sequential Line Search, Incremental Steepest Descent, GRG, Etc. 447
- 30.6.2 Local Surface Characterization Based: Newton–Raphson, Levenberg–Marquardt, Successive Quadratic, RLM, Quasi-Newton, Etc. 448
- 30.6.3 Direct Search with Single Trial Solution: Cyclic Heuristic, Hooke–Jeeves, and Nelder–Mead 448
- 30.6.4 Multiplayer Direct Search Optimizers: Leapfrogging, Particle Swarm, and Genetic Algorithms 448
- 30.7 Choices 448
- 30.8 Variable Classifications 449
- 30.8.1 Nominal 449
- 30.8.2 Ordinal 450
- 30.8.3 Cardinal 450
- 30.9 Constraints 451
- 30.10 Takeaway 453
- 30.11 Exercises 453

- 31 Response Surface Aberrations 459**
- 31.1 Introduction 459
- 31.2 Cliffs (Vertical Walls) 459
- 31.3 Sharp Valleys (or Ridges) 459
- 31.4 Striations 463
- 31.5 Level Spots (Functions 1, 27, 73, 84) 463

| | | |
|--------------------------------------------------------|----------------------------------------------------------------------------------------------------|------------|
| 31.6 | Hard-to-Find Optimum | 466 |
| 31.7 | Infeasible Calculations | 468 |
| 31.8 | Uniform Minimum | 468 |
| 31.9 | Noise: Stochastic Response | 469 |
| 31.10 | Multiple Optima | 471 |
| 31.11 | Takeaway | 473 |
| 31.12 | Exercises | 473 |
| 32 | Identifying the Models, OF, DV, Convergence Criteria, and Constraints | 475 |
| 32.1 | Introduction | 475 |
| 32.2 | Evaluate the Results | 476 |
| 32.3 | Takeaway | 482 |
| 32.4 | Exercises | 482 |
| 33 | Evaluating Optimizers | 489 |
| 33.1 | Introduction | 489 |
| 33.2 | Challenges to Optimizers | 490 |
| 33.3 | Stakeholders | 490 |
| 33.4 | Metrics of Optimizer Performance | 490 |
| 33.5 | Designing an Experimental Test | 492 |
| 33.6 | Takeaway | 495 |
| 33.7 | Exercises | 496 |
| 34 | Troubleshooting Optimizers | 499 |
| 34.1 | Introduction | 499 |
| 34.2 | DV Values Do Not Change | 499 |
| 34.3 | Multiple DV* Values for the Same OF* Value | 499 |
| 34.4 | EXE Error | 500 |
| 34.5 | Extreme Values | 500 |
| 34.6 | DV* Is Dependent on Convergence Threshold | 500 |
| 34.7 | OF* Is Irreproducible | 501 |
| 34.8 | Concern over Results | 501 |
| 34.9 | CDF Features | 501 |
| 34.10 | Parameter Correlation | 502 |
| 34.11 | Multiple Equivalent Solutions | 504 |
| 34.12 | Takeaway | 504 |
| 34.13 | Exercises | 504 |
| Section 6 Analysis of Leapfrogging Optimization | | 505 |
| 35 | Analysis of Leapfrogging | 507 |
| 35.1 | Introduction | 507 |
| 35.2 | Balance in an Optimizer | 508 |
| 35.3 | Number of Initializations to be Confident That the Best Will Draw All Others to the Global Optimum | 510 |

- 35.3.1 Methodology 511
- 35.3.2 Experimental 512
- 35.3.3 Results 513
- 35.4 Leap-To Window Amplification Analysis 515
- 35.5 Analysis of α and M to Prevent Convergence on the Side of a Hill 519
- 35.6 Analysis of α and M to Minimize NOFE 521
- 35.7 Probability Distribution of Leap-Overs 522
- 35.7.1 Data 526
- 35.8 Takeaway 527
- 35.9 Exercises 528

Section 7 Case Studies 529

36 Case Study 1: Economic Optimization of a Pipe System 531

- 36.1 Process and Analysis 531
- 36.1.1 Deterministic Continuum Model 531
- 36.1.2 Deterministic Discontinuous Model 534
- 36.1.3 Stochastic Discontinuous Model 535
- 36.2 Exercises 536

37 Case Study 2: Queuing Study 539

- 37.1 The Process and Analysis 539
- 37.2 Exercises 541

38 Case Study 3: Retirement Study 543

- 38.1 The Process and Analysis 543
- 38.2 Exercises 550

39 Case Study 4: A Goddard Rocket Study 551

- 39.1 The Process and Analysis 551
- 39.2 Pre-Assignment Note 554
- 39.3 Exercises 555

40 Case Study 5: Reservoir 557

- 40.1 The Process and Analysis 557
- 40.2 Exercises 559

41 Case Study 6: Area Coverage 561

- 41.1 Description and Analysis 561
- 41.2 Exercises 562

42 Case Study 7: Approximating Series Solution to an ODE 565

- 42.1 Concepts and Analysis 565
- 42.2 Exercises 568

| | | |
|-------------------|-------------------------------------------------------------|------------|
| 43 | Case Study 8: Horizontal Tank Vapor–Liquid Separator | 571 |
| 43.1 | Description and Analysis | 571 |
| 43.2 | Exercises | 576 |
| 44 | Case Study 9: <i>In Vitro</i> Fertilization | 579 |
| 44.1 | Description and Analysis | 579 |
| 44.2 | Exercises | 583 |
| 45 | Case Study 10: Data Reconciliation | 585 |
| 45.1 | Description and Analysis | 585 |
| 45.2 | Exercises | 588 |
| | | |
| Section 8 | Appendices | 591 |
| | | |
| Appendix A | Mathematical Concepts and Procedures | 593 |
| Appendix B | Root Finding | 605 |
| Appendix C | Gaussian Elimination | 611 |
| Appendix D | Steady-State Identification in Noisy Signals | 621 |
| Appendix E | Optimization Challenge Problems (2-D and Single OF) | 635 |
| Appendix F | Brief on VBA Programming: Excel in Office 2013 | 709 |
| | | |
| Section 9 | References and Index | 717 |
| | | |
| | References and Additional Resources | 719 |
| | Index | 723 |

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, Pretica 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

| | |
|---------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>a posteriori</i> | A Latin term, indicating “after it has been done.” A choice made after event outcomes have been observed. |
| <i>a priori</i> | A Latin term, indicating “before doing it.” A choice made before the event, from earlier experience or understanding, not after observation. |
| <i>Bottom line</i> | The reveal of comprehensive issues. |
| <i>CDF(OF*)</i> | The cumulative distribution function is a useful tool for visualizing the probability of finding the global optimum and the certainty of its location. |
| <i>Constraints</i> | 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 |

| | |
|------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>Convergence and stopping criteria</i> | <p>constraint violation, which softens the base of the cliff with a curve, which converts the surface to one that is analytically tractable. Something has to indicate that either the optimum is found or the optimizer is hopelessly lost and needs to be stopped. <i>Convergence</i> 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. <i>Stopping</i> 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.</p> |
| <i>Convex</i> | <p>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.</p> |
| <i>Customer</i> | <p>A person, or entity, or group that has a legitimate claim related to desirability or undesirability of your application. Alternately termed a stakeholder.</p> |
| <i>Derivative evaluation</i> | <p>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.</p> |
| <i>DV (decision variables)</i> | <p>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</p> |

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

| | |
|-------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | <p>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.</p> |
| <i>Number of independent trials</i> | <p>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.</p> |
| <i>Maximum or minimum</i> | <p>Since $\text{MAX } J = \text{OF}$ provides the same DV^* as $\text{MIN } J = -\text{OF}$, 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.</p> |
| <i>Model</i> | <p>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.</p> |
| <i>Multiple objectives</i> | <p>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.</p> |
| <i>Non-convex</i> | <p>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.</p> |
| <i>OF (objective function)</i> | <p>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</p> |

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

| | |
|----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | <p>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.</p> |
| <i>Risk</i> | <p>A penalty for a possible undesired event. Risk is the probability of the undesired event times the cost or equivalent impact of the event.</p> |
| <i>Stakeholder</i> | <p>A person, or entity, or group that has a legitimate claim related to desirability or undesirability of your application. Alternately termed a customer.</p> |
| <i>Surface aberrations</i> | <p>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?</p> |

| | |
|----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>Surrogate functions</i> | These are useful for optimizer testing. They are simple to code, quick to compute, and reveal the issues that serious applications would present. |
| <i>TS (trial solution)</i> | 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. |
| <i>Uncertainty</i> | 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, \dots | model coefficients, scalar variables |
| $w, x, y,$ z, \dots | model variables, scalar variables |
| \tilde{y} | modeled response value (as opposed to the measured value) |
| \mathbf{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 = |
| $ x $ | absolute value of a scalar variable |
| $\ \mathbf{x}\ $ | 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

