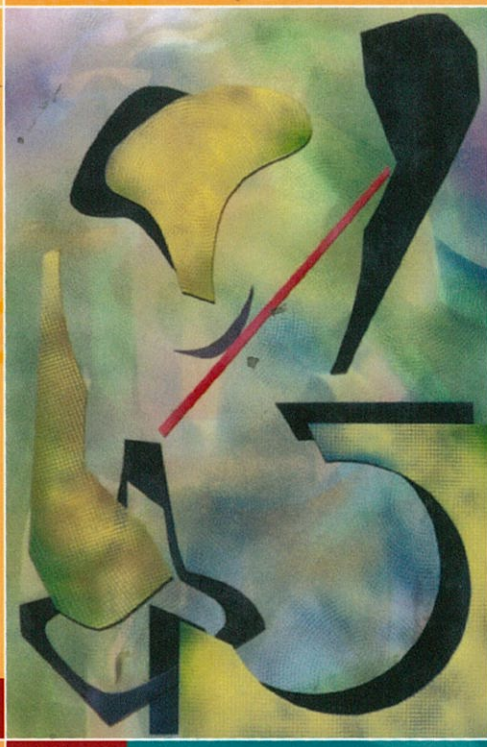
Zbigniew Michalewicz
David B. Fogel

How to Solve It: Modern Heuristics

Second Edition



Springer

Table of Contents

Introduction 1

I What Are the Ages of My Three Sons? 9

1 Why Are Some Problems Difficult to Solve? 11
1.1 The size of the search space 11
1.2 Modeling the problem 16
1.3 Change over time 19
1.4 Constraints 21
1.5 The problem of proving things 24
1.6 Your chance for glory 26
1.7 Summary 29

II How Important Is a Model? 31

2 Basic Concepts 35
2.1 Representation 35
2.2 The objective 36
2.3 The evaluation function 37
2.4 Defining a search problem 39
2.5 Neighborhoods and local optima 40
2.6 Hill-climbing methods 43
2.7 Can you sink this trick shot? 45
2.8 Summary 48

III What Are the Prices in 7–11? 49

3 Traditional Methods — Part 1 55
3.1 Exhaustive search 58
3.1.1 Enumerating the SAT 59
3.1.2 Enumerating the TSP 61
3.1.3 Enumerating the NLP 62
3.2 Local search 64
3.2.1 Local search and the SAT 65
3.2.2 Local search and the TSP 66

3.2.3 Local search and the NLP . . . 69
3.3 Linear programming: The simplex method . . . 76
3.4 Summary . . . 80

IV What Are the Numbers? . . . 83

4 Traditional Methods — Part 2 . . . 87
4.1 Greedy algorithms . . . 87
4.1.1 Greedy algorithms and the SAT . . . 87
4.1.2 Greedy algorithms and the TSP . . . 89
4.1.3 Greedy algorithms and the NLP . . . 90
4.2 Divide and conquer . . . 90
4.3 Dynamic programming . . . 93
4.4 Branch and bound . . . 101
4.5 A^* algorithm . . . 105
4.6 Summary . . . 109

V What's the Color of the Bear? . . . 111

5 Escaping Local Optima . . . 115
5.1 Simulated annealing . . . 117
5.2 Tabu search . . . 125
5.3 Summary . . . 134

VI How Good Is Your Intuition? . . . 135

6 An Evolutionary Approach . . . 139
6.1 Evolutionary approach for the SAT . . . 142
6.2 Evolutionary approach for the TSP . . . 145
6.3 Evolutionary approach for the NLP . . . 149
6.4 Summary . . . 151

VII One of These Things Is Not Like the Others . . . 157

7 Designing Evolutionary Algorithms . . . 161
7.1 Representation . . . 165
7.1.1 Fixed-length vectors of symbols . . . 166
7.1.2 Permutations . . . 167
7.1.3 Finite state machines . . . 168
7.1.4 Symbolic expressions . . . 168
7.2 Evaluation function . . . 169
7.3 Variation operators . . . 172
7.3.1 Fixed-length vectors of symbols . . . 173
7.3.2 Permutations . . . 174
7.3.3 Finite state machines . . . 176
7.3.4 Symbolic expressions . . . 177
7.4 Selection . . . 179

7.5 Initialization . . . 181
7.6 Summary . . . 183

VIII What Is the Shortest Way? . . . 185

8 The Traveling Salesman Problem . . . 189
8.1 In search of good variation operators . . . 191
8.2 Incorporating local search methods . . . 214
8.3 Other possibilities . . . 217
8.3.1 Edge assembly crossover . . . 217
8.3.2 Inver-over operator . . . 220
8.4 Summary . . . 223

IX Who Owns the Zebra? . . . 225

9 Constraint-Handling Techniques . . . 231
9.1 General considerations . . . 232
9.1.1 Designing $eval_f$ . . . 234
9.1.2 Design of $eval_u$ . . . 236
9.1.3 Relationship between $eval_f$ and $eval_u$ . . . 237
9.1.4 Rejecting infeasible solutions . . . 239
9.1.5 Repairing infeasible individuals . . . 239
9.1.6 Replacing individuals by their repaired versions . . . 240
9.1.7 Penalizing infeasible individuals . . . 240
9.1.8 Maintaining a feasible population using special representations and variation operators . . . 241
9.1.9 Using decoders . . . 242
9.1.10 Separating individuals and constraints . . . 243
9.1.11 Exploring boundaries between feasible and infeasible parts of the search space . . . 243
9.1.12 Finding feasible solutions . . . 244
9.2 Numerical optimization . . . 245
9.2.1 Methods based on preserving the feasibility of solutions . 246
9.2.2 Methods based on penalty functions . . . 249
9.2.3 Methods based on a search for feasible solutions . . . 257
9.2.4 Methods based on decoders . . . 264
9.2.5 Hybrid methods . . . 266
9.3 Summary . . . 268

X Can You Tune to the Problem? . . . 271

10 Tuning the Algorithm to the Problem . . . 277
10.1 Parameter control in evolutionary algorithms . . . 277
10.2 Illustrating the case with an NLP . . . 282
10.3 Taxonomy of control techniques . . . 284
10.4 The possibilities for parameter control . . . 288

10.4.1 Representation 288
10.4.2 Evaluation function 289
10.4.3 Mutation operators and their probabilities 290
10.4.4 Crossover operators and their probabilities 293
10.4.5 Parent selection 295
10.4.6 Population 295
10.5 Combining forms of parameter control 296
10.6 Summary 298

XI Can You Mate in Two Moves? 303

11 Time-Varying Environments and Noise 307
11.1 Life presents a dynamic landscape 307
11.2 The real world is noisy 317
11.2.1 Ensuring diversity 321
11.3 Modeling of ship trajectory 325
11.4 Summary 329

XII Day of the Week of January 1st 335

12 Neural Networks 339
12.1 Threshold neurons and linear discriminant functions 340
12.2 Back propagation for feed forward multilayer perceptrons 345
12.3 Training and testing 350
12.4 Recurrent networks and extended architectures 352
12.4.1 Standard recurrent network 352
12.4.2 Hopfield network 352
12.4.3 Boltzmann machine 354
12.4.4 Network of multiple interacting programs 354
12.5 Clustering with competitive networks 356
12.6 Using neural networks to solve the TSP 358
12.7 Evolving neural networks 360
12.8 Summary 361

XIII What Was the Length of the Rope? 363

13 Fuzzy Systems 367
13.1 Fuzzy sets 368
13.2 Fuzzy sets and probability measures 368
13.3 Fuzzy set operations 369
13.4 Fuzzy relationships 372
13.5 Designing a fuzzy controller 375
13.6 Fuzzy clustering 380
13.7 Fuzzy neural networks 382
13.8 A fuzzy TSP 385
13.9 Evolving fuzzy systems 387

13.10 Summary 388

XIV Everything Depends on Something Else 389

14 Coevolutionary Systems 397
14.1 Playing a game 399
14.2 Learning to play checkers 400
14.3 Optimization through competing populations 403
14.4 Another example with a game 408
14.5 Artificial life 411
14.6 Games of cooperation and competition 412
14.7 Modeling intelligent agents 418
14.8 Issues in coevolution 420
14.9 An example of cooperative coevolution on the TSP 424
14.10 Summary 430

XV Who's Taller? 431

15 Multicriteria Decision-Making 435
15.1 Single numeric approach 440
15.1.1 The Valuated State Space approach 443
15.1.2 Fuzzy multicriteria decision-making 449
15.1.3 Summary 450
15.2 Evolutionary multiobjective optimization approach 451
15.2.1 Brief history 452
15.2.2 Evolutionary algorithm for multiobjective NLP 458
15.3 Summary 462

XVI Do You Like Simple Solutions? 465

16 Hybrid Systems 471
16.1 Summary 480

17 Summary 483

Appendix A: Probability and Statistics 495
A.1 Basic concepts of probability 495
A.2 Random variables 497
A.2.1 Discrete random variables 497
A.2.2 Continuous random variables 500
A.3 Descriptive statistics of random variables 502
A.4 Limit theorems and inequalities 505
A.5 Adding random variables 506
A.6 Generating random numbers on a computer 507
A.7 Estimation 508
A.8 Statistical hypothesis testing 511
A.9 Linear regression 512

A.10 Summary 514

Appendix B: Problems and Projects 515
B.1 Trying some practical problems 517
B.2 Reporting computational experiments with heuristic methods . . 522

References 525

Index 551