METAHEURISTICs
FROM DESIGN TO IMPLEMENTATION

El-Ghazali Talbi
University of Lille – CNRS – INRIA
## CONTENTS

<table>
<thead>
<tr>
<th>Preface</th>
<th>xvii</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acknowledgments</td>
<td>xxiii</td>
</tr>
<tr>
<td>Glossary</td>
<td>xxv</td>
</tr>
</tbody>
</table>

### 1 Common Concepts for Metaheuristics

1.1 Optimization Models
   1.1.1 Classical Optimization Models
   1.1.2 Complexity Theory
      1.1.2.1 Complexity of Algorithms
      1.1.2.2 Complexity of Problems
   1.1.3 Other Models for Optimization
      1.1.3.1 Optimization Under Uncertainty
      1.1.3.2 Dynamic Optimization
      1.1.3.2.1 Multiperiodic Optimization
      1.1.3.3 Robust Optimization
   1.1.4 Optimization Methods
      1.1.4.1 Exact Methods
      1.1.4.2 Approximate Algorithms
      1.1.4.2.1 Approximation Algorithms
      1.1.4.3 Metaheuristics
      1.1.4.4 Greedy Algorithms
      1.1.4.5 When Using Metaheuristics?
   1.1.5 Main Common Concepts for Metaheuristics
      1.1.5.1 Representation
      1.1.5.1.1 Linear Representations
      1.1.5.1.2 Nonlinear Representations
      1.1.5.1.3 Representation-Solution Mapping
      1.1.5.1.4 Direct Versus Indirect Encodings
      1.1.5.2 Objective Function
      1.1.5.2.1 Self-Sufficient Objective Functions

1.2 Other Models for Optimization
   1.2.1 Optimization Under Uncertainty
   1.2.2 Dynamic Optimization
   1.2.2.1 Multiperiodic Optimization
   1.2.3 Robust Optimization

1.3 Optimization Methods
   1.3.1 Exact Methods
   1.3.2 Approximate Algorithms
   1.3.2.1 Approximation Algorithms
   1.3.3 Metaheuristics
   1.3.4 Greedy Algorithms
   1.3.5 When Using Metaheuristics?

1.4 Main Common Concepts for Metaheuristics
   1.4.1 Representation
      1.4.1.1 Linear Representations
      1.4.1.2 Nonlinear Representations
      1.4.1.3 Representation-Solution Mapping
      1.4.1.4 Direct Versus Indirect Encodings
   1.4.2 Objective Function
      1.4.2.1 Self-Sufficient Objective Functions

1.5 When Using Metaheuristics?

1.6 Main Common Concepts for Metaheuristics

1.7 When Using Metaheuristics?

1.8 Main Common Concepts for Metaheuristics

1.9 When Using Metaheuristics?

1.10 Main Common Concepts for Metaheuristics

1.11 When Using Metaheuristics?

1.12 Main Common Concepts for Metaheuristics

1.13 When Using Metaheuristics?

1.14 Main Common Concepts for Metaheuristics

1.15 When Using Metaheuristics?
1.4.2.2 Guiding Objective Functions 44
1.4.2.3 Representation Decoding 45
1.4.2.4 Interactive Optimization 46
1.4.2.5 Relative and Competitive Objective Functions 47
1.4.2.6 Meta-Modeling 47

1.5 Constraint Handling 48
1.5.1 Reject Strategies 49
1.5.2 Penalizing Strategies 49
1.5.3 Repairing Strategies 52
1.5.4 Decoding Strategies 53
1.5.5 Preserving Strategies 53

1.6 Parameter Tuning 54
1.6.1 Off-Line Parameter Initialization 54
1.6.2 Online Parameter Initialization 56

1.7 Performance Analysis of Metaheuristics 57
1.7.1 Experimental Design 57
1.7.2 Measurement 60
1.7.2.1 Quality of Solutions 60
1.7.2.2 Computational Effort 62
1.7.2.3 Robustness 62
1.7.2.4 Statistical Analysis 63
1.7.2.5 Ordinal Data Analysis 64
1.7.3 Reporting 65

1.8 Software Frameworks for Metaheuristics 67
1.8.1 Why a Software Framework for Metaheuristics? 67
1.8.2 Main Characteristics of Software Frameworks 69
1.8.3 ParadisEO Framework 71
1.8.3.1 ParadisEO Architecture 74

1.9 Conclusions 76

1.10 Exercises 79

2 Single-Solution Based Metaheuristics 87

2.1 Common Concepts for Single-Solution Based Metaheuristics 87
2.1.1 Neighborhood 88
2.1.2 Very Large Neighborhoods 94
2.1.2.1 Heuristic Search in Large Neighborhoods 95
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1.2.2</td>
<td>Exact Search in Large Neighborhoods</td>
<td>98</td>
</tr>
<tr>
<td>2.1.2.3</td>
<td>Polynomial-Specific Neighborhoods</td>
<td>100</td>
</tr>
<tr>
<td>2.1.3</td>
<td>Initial Solution</td>
<td>101</td>
</tr>
<tr>
<td>2.1.4</td>
<td>Incremental Evaluation of the Neighborhood</td>
<td>102</td>
</tr>
<tr>
<td>2.2</td>
<td>Fitness Landscape Analysis</td>
<td>103</td>
</tr>
<tr>
<td>2.2.1</td>
<td>Distances in the Search Space</td>
<td>106</td>
</tr>
<tr>
<td>2.2.2</td>
<td>Landscape Properties</td>
<td>108</td>
</tr>
<tr>
<td>2.2.2.1</td>
<td>Distribution Measures</td>
<td>109</td>
</tr>
<tr>
<td>2.2.2.2</td>
<td>Correlation Measures</td>
<td>111</td>
</tr>
<tr>
<td>2.2.3</td>
<td>Breaking Plateaus in a Flat Landscape</td>
<td>119</td>
</tr>
<tr>
<td>2.3</td>
<td>Local Search</td>
<td>121</td>
</tr>
<tr>
<td>2.3.1</td>
<td>Selection of the Neighbor</td>
<td>123</td>
</tr>
<tr>
<td>2.3.2</td>
<td>Escaping from Local Optima</td>
<td>125</td>
</tr>
<tr>
<td>2.4</td>
<td>Simulated Annealing</td>
<td>126</td>
</tr>
<tr>
<td>2.4.1</td>
<td>Move Acceptance</td>
<td>129</td>
</tr>
<tr>
<td>2.4.2</td>
<td>Cooling Schedule</td>
<td>130</td>
</tr>
<tr>
<td>2.4.2.1</td>
<td>Initial Temperature</td>
<td>130</td>
</tr>
<tr>
<td>2.4.2.2</td>
<td>Equilibrium State</td>
<td>131</td>
</tr>
<tr>
<td>2.4.2.3</td>
<td>Cooling</td>
<td>131</td>
</tr>
<tr>
<td>2.4.2.4</td>
<td>Stopping Condition</td>
<td>133</td>
</tr>
<tr>
<td>2.4.3</td>
<td>Other Similar Methods</td>
<td>133</td>
</tr>
<tr>
<td>2.4.3.1</td>
<td>Threshold Accepting</td>
<td>133</td>
</tr>
<tr>
<td>2.4.3.2</td>
<td>Record-to-Record Travel</td>
<td>137</td>
</tr>
<tr>
<td>2.4.3.3</td>
<td>Great Deluge Algorithm</td>
<td>137</td>
</tr>
<tr>
<td>2.4.3.4</td>
<td>Demon Algorithms</td>
<td>138</td>
</tr>
<tr>
<td>2.5</td>
<td>Tabu Search</td>
<td>140</td>
</tr>
<tr>
<td>2.5.1</td>
<td>Short-Term Memory</td>
<td>142</td>
</tr>
<tr>
<td>2.5.2</td>
<td>Medium-Term Memory</td>
<td>144</td>
</tr>
<tr>
<td>2.5.3</td>
<td>Long-Term Memory</td>
<td>145</td>
</tr>
<tr>
<td>2.6</td>
<td>Iterated Local Search</td>
<td>146</td>
</tr>
<tr>
<td>2.6.1</td>
<td>Perturbation Method</td>
<td>148</td>
</tr>
<tr>
<td>2.6.2</td>
<td>Acceptance Criteria</td>
<td>149</td>
</tr>
<tr>
<td>2.7</td>
<td>Variable Neighborhood Search</td>
<td>150</td>
</tr>
<tr>
<td>2.7.1</td>
<td>Variable Neighborhood Descent</td>
<td>150</td>
</tr>
<tr>
<td>2.7.2</td>
<td>General Variable Neighborhood Search</td>
<td>151</td>
</tr>
<tr>
<td>2.8</td>
<td>Guided Local Search</td>
<td>154</td>
</tr>
</tbody>
</table>
2.9 Other Single-Solution Based Metaheuristics 157
  2.9.1 Smoothing Methods 157
  2.9.2 Noisy Method 160
  2.9.3 GRASP 164

2.10 S-Metaheuristic Implementation Under ParadisEO 168
  2.10.1 Common Templates for Metaheuristics 169
  2.10.2 Common Templates for S-Metaheuristics 170
  2.10.3 Local Search Template 170
  2.10.4 Simulated Annealing Template 172
  2.10.5 Tabu Search Template 173
  2.10.6 Iterated Local Search Template 175

2.11 Conclusions 177

2.12 Exercises 180

3 Population-Based Metaheuristics 190

  3.1 Common Concepts for Population-Based Metaheuristics 191
    3.1.1 Initial Population 193
      3.1.1.1 Random Generation 194
      3.1.1.2 Sequential Diversification 195
      3.1.1.3 Parallel Diversification 195
      3.1.1.4 Heuristic Initialization 198
    3.1.2 Stopping Criteria 198

  3.2 Evolutionary Algorithms 199
    3.2.1 Genetic Algorithms 201
    3.2.2 Evolution Strategies 202
    3.2.3 Evolutionary Programming 203
    3.2.4 Genetic Programming 203

  3.3 Common Concepts for Evolutionary Algorithms 205
    3.3.1 Selection Methods 206
      3.3.1.1 Roulette Wheel Selection 206
      3.3.1.2 Stochastic Universal Sampling 206
      3.3.1.3 Tournament Selection 207
      3.3.1.4 Rank-Based Selection 207
    3.3.2 Reproduction 208
      3.3.2.1 Mutation 208
      3.3.2.2 Recombination or Crossover 213
    3.3.3 Replacement Strategies 221
3.4 Other Evolutionary Algorithms 221
  3.4.1 Estimation of Distribution Algorithms 222
  3.4.2 Differential Evolution 225
  3.4.3 Coevolutionary Algorithms 228
  3.4.4 Cultural Algorithms 232

3.5 Scatter Search 233
  3.5.1 Path Relinking 237

3.6 Swarm Intelligence 240
  3.6.1 Ant Colony Optimization Algorithms 240
    3.6.1.1 ACO for Continuous Optimization Problems 247
  3.6.2 Particle Swarm Optimization 247
    3.6.2.1 Particles Neighborhood 248
    3.6.2.2 PSO for Discrete Problems 252

3.7 Other Population-Based Methods 255
  3.7.1 Bees Colony 255
    3.7.1.1 Bees in Nature 255
    3.7.1.2 Nest Site Selection 256
    3.7.1.3 Food Foraging 257
    3.7.1.4 Marriage Process 262
  3.7.2 Artificial Immune Systems 264
    3.7.2.1 Natural Immune System 264
    3.7.2.2 Clonal Selection Theory 265
    3.7.2.3 Negative Selection Principle 268
    3.7.2.4 Immune Network Theory 268
    3.7.2.5 Danger Theory 269

3.8 P-metaheuristics Implementation Under ParadisEO 270
  3.8.1 Common Components and Programming Hints 270
    3.8.1.1 Main Core Templates—ParadisEO–EO’s Functors 270
    3.8.1.2 Representation 272
  3.8.2 Fitness Function 274
    3.8.2.1 Initialization 274
    3.8.2.2 Stopping Criteria, Checkpoints, and Statistics 275
    3.8.2.3 Dynamic Parameter Management and State Loader/Register 277
  3.8.3 Evolutionary Algorithms Under ParadisEO 278
    3.8.3.1 Representation 278
    3.8.3.2 Initialization 279
    3.8.3.3 Evaluation 279
CONTENTS

3.8.3.4 Variation Operators 279
3.8.3.5 Evolution Engine 283
3.8.3.6 Evolutionary Algorithms 285
3.8.4 Particle Swarm Optimization Under ParadisEO 286
   3.8.4.1 Illustrative Example 292
3.8.5 Estimation of Distribution Algorithm Under ParadisEO 293
3.9 Conclusions 294
3.10 Exercises 296

4 Metaheuristics for Multiobjective Optimization 308
4.1 Multiobjective Optimization Concepts 310
4.2 Multiobjective Optimization Problems 315
   4.2.1 Academic Applications 316
      4.2.1.1 Multiobjective Continuous Problems 316
      4.2.1.2 Multiobjective Combinatorial Problems 317
   4.2.2 Real-Life Applications 318
   4.2.3 Multicriteria Decision Making 320
4.3 Main Design Issues of Multiobjective Metaheuristics 322
4.4 Fitness Assignment Strategies 323
   4.4.1 Scalar Approaches 324
      4.4.1.1 Aggregation Method 324
      4.4.1.2 Weighted Metrics 327
      4.4.1.3 Goal Programming 330
      4.4.1.4 Achievement Functions 330
      4.4.1.5 Goal Attainment 330
      4.4.1.6 $\epsilon$-Constraint Method 332
   4.4.2 Criterion-Based Methods 334
      4.4.2.1 Parallel Approach 334
      4.4.2.2 Sequential or Lexicographic Approach 335
   4.4.3 Dominance-Based Approaches 337
   4.4.4 Indicator-Based Approaches 341
4.5 Diversity Preservation 343
   4.5.1 Kernel Methods 344
   4.5.2 Nearest-Neighbor Methods 346
   4.5.3 Histograms 347
4.6 Elitism 347
4.7 Performance Evaluation and Pareto Front Structure 350
4.7.1 Performance Indicators 350
   4.7.1.1 Convergence-Based Indicators 352
   4.7.1.2 Diversity-Based Indicators 354
   4.7.1.3 Hybrid Indicators 355
4.7.2 Landscape Analysis of Pareto Structures 358
4.8 Multiobjective Metaheuristics Under ParadisEO 361
   4.8.1 Software Frameworks for Multiobjective Metaheuristics 362
   4.8.2 Common Components 363
      4.8.2.1 Representation 363
      4.8.2.2 Fitness Assignment Schemes 364
      4.8.2.3 Diversity Assignment Schemes 366
      4.8.2.4 Elitism 367
      4.8.2.5 Statistical Tools 367
   4.8.3 Multiobjective EAs-Related Components 368
      4.8.3.1 Selection Schemes 369
      4.8.3.2 Replacement Schemes 370
      4.8.3.3 Multiobjective Evolutionary Algorithms 371
4.9 Conclusions and Perspectives 373
4.10 Exercises 375

5 Hybrid Metaheuristics 385
   5.1 Hybrid Metaheuristics 386
      5.1.1 Design Issues 386
         5.1.1.1 Hierarchical Classification 386
         5.1.1.2 Flat Classification 394
      5.1.2 Implementation Issues 399
         5.1.2.1 Dedicated Versus General-Purpose Computers 399
         5.1.2.2 Sequential Versus Parallel 399
      5.1.3 A Grammar for Extended Hybridization Schemes 400
   5.2 Combining Metaheuristics with Mathematical Programming 401
      5.2.1 Mathematical Programming Approaches 402
         5.2.1.1 Enumerative Algorithms 402
         5.2.1.2 Relaxation and Decomposition Methods 405
         5.2.1.3 Branch and Cut and Price Algorithms 407
      5.2.2 Classical Hybrid Approaches 407
         5.2.2.1 Low-Level Relay Hybrids 408
         5.2.2.2 Low-Level Teamwork Hybrids 411
5.2.2.3 High-Level Relay Hybrids 413
5.2.2.4 High-Level Teamwork Hybrids 416

5.3 Combining Metaheuristics with Constraint Programming 418
5.3.1 Constraint Programming 418
5.3.2 Classical Hybrid Approaches 419
  5.3.2.1 Low-Level Relay Hybrids 420
  5.3.2.2 Low-Level Teamwork Hybrids 420
  5.3.2.3 High-Level Relay Hybrids 422
  5.3.2.4 High-Level Teamwork Hybrids 422

5.4 Hybrid Metaheuristics with Machine Learning and Data Mining 423
5.4.1 Data Mining Techniques 423
5.4.2 Main Schemes of Hybridization 425
  5.4.2.1 Low-Level Relay Hybrid 425
  5.4.2.2 Low-Level Teamwork Hybrids 426
  5.4.2.3 High-Level Relay Hybrid 428
  5.4.2.4 High-Level Teamwork Hybrid 431

5.5 Hybrid Metaheuristics for Multiobjective Optimization 432
5.5.1 Combining Metaheuristics for MOPs 432
  5.5.1.1 Low-Level Relay Hybrids 432
  5.5.1.2 Low-Level Teamwork Hybrids 433
  5.5.1.3 High-Level Relay Hybrids 434
  5.5.1.4 High-Level Teamwork Hybrid 436
5.5.2 Combining Metaheuristics with Exact Methods for MOP 438
5.5.3 Combining Metaheuristics with Data Mining for MOP 444

5.6 Hybrid Metaheuristics Under ParadisEO 448
5.6.1 Low-Level Hybrids Under ParadisEO 448
5.6.2 High-Level Hybrids Under ParadisEO 451
5.6.3 Coupling with Exact Algorithms 451

5.7 Conclusions and Perspectives 452

5.8 Exercises 454

6 Parallel Metaheuristics 460
6.1 Parallel Design of Metaheuristics 462
  6.1.1 Algorithmic-Level Parallel Model 463
    6.1.1.1 Independent Algorithmic-Level Parallel Model 463
    6.1.1.2 Cooperative Algorithmic-Level Parallel Model 465
Appendix: UML and C++

A.1 A Brief Overview of UML Notations 535
A.2 A Brief Overview of the C++ Template Concept 536

References 539

Index 587