

1.4.2.2	<i>Guiding Objective Functions</i>	44
1.4.2.3	<i>Representation Decoding</i>	45
1.4.2.4	<i>Interactive Optimization</i>	46
1.4.2.5	<i>Relative and Competitive Objective Functions</i>	47
1.4.2.6	<i>Meta-Modeling</i>	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	<i>Quality of Solutions</i>	60
1.7.2.2	<i>Computational Effort</i>	62
1.7.2.3	<i>Robustness</i>	62
1.7.2.4	<i>Statistical Analysis</i>	63
1.7.2.5	<i>Ordinal Data Analysis</i>	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	<i>ParadisEO Architecture</i>	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	<i>Heuristic Search in Large Neighborhoods</i>	95

2.1.2.2	<i>Exact Search in Large Neighborhoods</i>	98
2.1.2.3	<i>Polynomial-Specific Neighborhoods</i>	100
2.1.3	Initial Solution	101
2.1.4	Incremental Evaluation of the Neighborhood	102
2.2	Fitness Landscape Analysis	103
2.2.1	Distances in the Search Space	106
2.2.2	Landscape Properties	108
2.2.2.1	<i>Distribution Measures</i>	109
2.2.2.2	<i>Correlation Measures</i>	111
2.2.3	Breaking Plateaus in a Flat Landscape	119
2.3	Local Search	121
2.3.1	Selection of the Neighbor	123
2.3.2	Escaping from Local Optima	125
2.4	Simulated Annealing	126
2.4.1	Move Acceptance	129
2.4.2	Cooling Schedule	130
2.4.2.1	<i>Initial Temperature</i>	130
2.4.2.2	<i>Equilibrium State</i>	131
2.4.2.3	<i>Cooling</i>	131
2.4.2.4	<i>Stopping Condition</i>	133
2.4.3	Other Similar Methods	133
2.4.3.1	<i>Threshold Accepting</i>	133
2.4.3.2	<i>Record-to-Record Travel</i>	137
2.4.3.3	<i>Great Deluge Algorithm</i>	137
2.4.3.4	<i>Demon Algorithms</i>	138
2.5	Tabu Search	140
2.5.1	Short-Term Memory	142
2.5.2	Medium-Term Memory	144
2.5.3	Long-Term Memory	145
2.6	Iterated Local Search	146
2.6.1	Perturbation Method	148
2.6.2	Acceptance Criteria	149
2.7	Variable Neighborhood Search	150
2.7.1	Variable Neighborhood Descent	150
2.7.2	General Variable Neighborhood Search	151
2.8	Guided Local Search	154

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

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	ϵ -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	<i>High-Level Relay Hybrids</i>	413	6.1.2	Iteration-Level Parallel Model	471
5.2.2.4	<i>High-Level Teamwork Hybrids</i>	416	6.1.2.1	<i>Iteration-Level Model for S-Metaheuristics</i>	471
5.3	Combining Metaheuristics with Constraint Programming	418	6.1.2.2	<i>Iteration-Level Model for P-Metaheuristics</i>	472
5.3.1	Constraint Programming	418	6.1.3	Solution-Level Parallel Model	476
5.3.2	Classical Hybrid Approaches	419	6.1.4	Hierarchical Combination of the Parallel Models	478
5.3.2.1	<i>Low-Level Relay Hybrids</i>	420	6.2	Parallel Implementation of Metaheuristics	478
5.3.2.2	<i>Low-Level Teamwork Hybrids</i>	420	6.2.1	Parallel and Distributed Architectures	480
5.3.2.3	<i>High-Level Relay Hybrids</i>	422	6.2.2	Dedicated Architectures	486
5.3.2.4	<i>High-Level Teamwork Hybrids</i>	422	6.2.3	Parallel Programming Environments and Middlewares	488
5.4	Hybrid Metaheuristics with Machine Learning and Data Mining	423	6.2.4	Performance Evaluation	493
5.4.1	Data Mining Techniques	423	6.2.5	Main Properties of Parallel Metaheuristics	496
5.4.2	Main Schemes of Hybridization	425	6.2.6	Algorithmic-Level Parallel Model	498
5.4.2.1	<i>Low-Level Relay Hybrid</i>	425	6.2.7	Iteration-Level Parallel Model	500
5.4.2.2	<i>Low-Level Teamwork Hybrids</i>	426	6.2.8	Solution-Level Parallel Model	502
5.4.2.3	<i>High-Level Relay Hybrid</i>	428	6.3	Parallel Metaheuristics for Multiobjective Optimization	504
5.4.2.4	<i>High-Level Teamwork Hybrid</i>	431	6.3.1	Algorithmic-Level Parallel Model for MOP	505
5.5	Hybrid Metaheuristics for Multiobjective Optimization	432	6.3.2	Iteration-Level Parallel Model for MOP	507
5.5.1	Combining Metaheuristics for MOPs	432	6.3.3	Solution-Level Parallel Model for MOP	507
5.5.1.1	<i>Low-Level Relay Hybrids</i>	432	6.3.4	Hierarchical Parallel Model for MOP	509
5.5.1.2	<i>Low-Level Teamwork Hybrids</i>	433	6.4	Parallel Metaheuristics Under ParadisEO	512
5.5.1.3	<i>High-Level Relay Hybrids</i>	434	6.4.1	Parallel Frameworks for Metaheuristics	512
5.5.1.4	<i>High-Level Teamwork Hybrid</i>	436	6.4.2	Design of Algorithmic-Level Parallel Models	513
5.5.2	Combining Metaheuristics with Exact Methods for MOP	438	6.4.2.1	<i>Algorithms and Transferred Data (What?)</i>	514
5.5.3	Combining Metaheuristics with Data Mining for MOP	444	6.4.2.2	<i>Transfer Control (When?)</i>	514
5.6	Hybrid Metaheuristics Under ParadisEO	448	6.4.2.3	<i>Exchange Topology (Where?)</i>	515
5.6.1	Low-Level Hybrids Under ParadisEO	448	6.4.2.4	<i>Replacement Strategy (How?)</i>	517
5.6.2	High-Level Hybrids Under ParadisEO	451	6.4.2.5	<i>Parallel Implementation</i>	517
5.6.3	Coupling with Exact Algorithms	451	6.4.2.6	<i>A Generic Example</i>	518
5.7	Conclusions and Perspectives	452	6.4.2.7	<i>Island Model of EAs Within ParadisEO</i>	519
5.8	Exercises	454	6.4.3	Design of Iteration-Level Parallel Models	521
6	Parallel Metaheuristics	460	6.4.3.1	<i>The Generic Multistart Paradigm</i>	521
6.1	Parallel Design of Metaheuristics	462	6.4.3.2	<i>Use of the Iteration-Level Model</i>	523
6.1.1	Algorithmic-Level Parallel Model	463	6.4.4	Design of Solution-Level Parallel Models	524
6.1.1.1	<i>Independent Algorithmic-Level Parallel Model</i>	463	6.4.5	Implementation of Sequential Metaheuristics	524
6.1.1.2	<i>Cooperative Algorithmic-Level Parallel Model</i>	465	6.4.6	Implementation of Parallel and Distributed Algorithms	525
			6.4.7	Deployment of ParadisEO-PEO	528
			6.5	Conclusions and Perspectives	529
			6.6	Exercises	531

Appendix: UML and C++	535
A.1 A Brief Overview of UML Notations	535
A.2 A Brief Overview of the C++ Template Concept	536
References	539
Index	587