Contents

Foreword **vii**

Preface **xi**

1 Brief History of Test Theory and Design **1**
1.1 Classical Test Design 2
1.1.1 Standardized Testing in Psychology 2
1.1.2 Classical Test Theory 4
1.1.3 Discussion 8
1.2 Modern Test Design 9
1.2.1 New Notion of Standardization 9
1.2.2 Item-Response Theory 11
1.2.3 Item Calibration and Ability Measurement 14
1.2.4 Test and Item Information Functions 16
1.2.5 Test Characteristic Function 17
1.2.6 Comparison Between Classical and IRT Parameters 19
1.2.7 Ability Scale and Item Mapping 19
1.2.8 Birnbaum Approach to Test Design 21
1.3 Test Design in This Book 23
1.3.1 Four Modes of Test Assembly 24
1.3.2 Choice of Test Assembly Modes 26
1.4 An Application of Integer Programming to Test Assembly 26

1.5 Literature 28
1.6 Summary 30
1.7 Exercises 31

2 Formulating Test Specifications 33
2.1 Examples of Test Specifications 34
2.2 Classification of Attributes 36
2.2.1 Type of Attribute 36
2.2.2 Level of Attribute 37
2.3 Constraints and Objectives 38
2.4 Standard Form of the Set of Test Specifications 40
2.4.1 Number of Objective Functions 40
2.4.2 Number of Constraints 41
2.5 Literature 42
2.6 Summary 43
2.7 Exercises 44

3 Modeling Test-Assembly Problems 47
3.1 Identifying Decision Variables 48
3.2 Modeling Constraints 51
3.2.1 Quantitative Constraints 51
3.2.2 Categorical Constraints 55
3.2.3 Logical Constraints 59
3.2.4 Checking Constraints 61
3.3 Formulating Objective Functions 64
3.3.1 Quantitative Objective Functions 64
3.3.2 Categorical Objective Functions 66
3.3.3 Objective Functions with Goal Values 67
3.3.4 Multiobjective Test Assembly 68
3.3.5 Nonlinear Objectives 72
3.4 Literature 72
3.5 Summary 72
3.6 Exercises 74

4 Solving Test-Assembly Problems 77
4.1 Standard Model for a Single Test 78
4.1.1 Checking Interactions Between the Objective Function and Constraints 80
4.2 Branch-and-Bound Search 81
4.2.1 Tree Search 82
4.2.2 Implementation Decisions 84
4.2.3 Problem Size and Solution Time 85
4.2.4 A Useful Approximation 86
4.2.5 Software 87
4.3 Network-Flow Approximation 87

4.4 Constructive Heuristics . . . 89
4.4.1 Greedy Heuristics . . . 89
4.4.2 Luecht Heuristic . . . 91
4.4.3 Swanson-Stocking Heuristic . . . 91
4.5 Local Search Heuristics . . . 92
4.5.1 Genetic Algorithms . . . 93
4.5.2 Simulated Annealing . . . 94
4.6 Simultaneous and Sequential Optimization . . . 96
4.7 Optimal Design Approach . . . 98
4.8 Literature . . . 101
4.9 Summary . . . 102

5 Models for Assembling Single Tests 105
5.1 IRT-Based Test Assembly . . . 106
5.1.1 Absolute and Relative Targets . . . 107
5.1.2 Methods for Specifying Targets for Information Functions . . . 108
5.1.3 Assembling Tests for Absolute Targets . . . 110
5.1.4 Assembling Tests for Relative Targets . . . 113
5.1.5 Cutoff Scores . . . 114
5.1.6 Empirical Examples . . . 114
5.2 Classical Test Assembly . . . 115
5.2.1 Maximizing Test Reliability . . . 117
5.2.2 Maximizing Predictive Validity . . . 118
5.2.3 Constraining Test Reliability . . . 119
5.2.4 Empirical Example . . . 120
5.3 Matching Observed-Score Distributions . . . 121
5.3.1 Conditions on the Response Functions . . . 122
5.3.2 Constraints in the Test-Assembly Model . . . 123
5.3.3 Discussion . . . 124
5.3.4 Empirical Examples . . . 124
5.4 Item Matching . . . 125
5.4.1 Matching Items in a Reference Test . . . 129
5.4.2 Test Splitting . . . 131
5.4.3 Discussion . . . 133
5.4.4 Empirical Example . . . 133
5.5 Literature . . . 135
5.6 Summary . . . 136
5.7 Exercises . . . 137

6 Models for Assembling Multiple Tests 139
6.1 Sequential Assembly . . . 141
6.1.1 Heuristic Correction . . . 142
6.2 Simultaneous Assembly . . . 142
6.2.1 Item Overlap . . . 144

6.2.2 Controlling Targets Through Constraints 145
6.3 Big-Shadow-Test Method 146
6.3.1 Discussion 150
6.4 Alternative Backup Methods 151
6.5 Optimizing BIB Designs 152
6.6 Empirical Examples 155
6.7 Literature 159
6.8 Summary 161
6.9 Exercises 162

7 Models for Assembling Tests with Item Sets 165
7.1 Simultaneous Selection of Items and Stimuli 166
7.2 Power-Set Method 170
7.3 Edited-Set Method 174
7.4 Pivot-Item Method 174
7.5 Two-Stage Method 175
7.5.1 Stage 1: Selection of Stimuli 175
7.5.2 Stage 2: Selection of Items from Sets 178
7.5.3 Alternative Version 179
7.6 Empirical Example 179
7.7 Literature 185
7.8 Summary 185
7.9 Exercises 186

8 Models for Assembling Tests Measuring Multiple Abilities 189
8.1 Different Cases of Multidimensional Testing 190
8.1.1 Both Abilities Intentional 190
8.1.2 One Nuisance Ability 191
8.1.3 Composite Ability 191
8.1.4 Simple Structure of Multidimensional Abilities . . 192
8.1.5 Simple Structure of Unidimensional Abilities . . . 192
8.2 Variance Functions 192
8.3 Linearization of the Problem 194
8.3.1 Linear Decomposition 194
8.3.2 Linear Approximation 197
8.4 Main Models 197
8.4.1 Model for Relative Targets 198
8.4.2 Model for Absolute Targets 200
8.4.3 Applications to Different Cases 200
8.5 Alternative Objectives for Multidimensional Test Assembly 203
8.5.1 Matching Observed-Score Distributions 203
8.5.2 Item Matching 204
8.5.3 Other Generalizations of Unidimensional Problems 204

8.6 Empirical Example 204
8.7 Literature 207
8.8 Summary 207
8.9 Exercises 209

9 Models for Adaptive Test Assembly 211
9.1 Shadow-Test Approach 213
9.1.1 Random Test Length 214
9.1.2 Fixed Test Length 214
9.1.3 Definition of Shadow Tests 216
9.1.4 Standard Model for a Shadow Test 217
9.1.5 Calculating Shadow Tests 218
9.1.6 Empirical Example 219
9.1.7 Discussion 221
9.2 Alternative Objective Functions 222
9.2.1 Kullback-Leibler Information 223
9.2.2 Bayesian Item-Selection Criteria 223
9.3 Adaptive Testing with Item Sets 224
9.4 Controlling Item Exposure 225
9.4.1 Alpha Stratification 225
9.4.2 Sympson-Hetter Method 229
9.4.3 Multiple-Shadow-Test Approach 230
9.4.4 Method with Ineligibility Constraints 233
9.5 Controlling the Speededness of the Test 235
9.5.1 Response-Time Model 237
9.5.2 Ability and Speed as Intentional Factors 238
9.5.3 Speed as a Nuisance Factor 239
9.6 Reporting Scores on a Reference Test 241
9.7 Multidimensional Adaptive Test Assembly 248
9.7.1 Minimizing Error Variances 248
9.7.2 Computational Aspects 251
9.7.3 Maximizing Kullback-Leibler Information 251
9.7.4 Empirical Examples 252
9.8 Final Comments 253
9.9 Literature 257
9.10 Summary 259
9.11 Exercises 261

10 Designing Item Pools for Programs with Fixed Tests 265
10.1 Definition of Design Space 266
10.2 Programs with Parallel Forms of a Single Test 268
10.2.1 Standard Design Model 268
10.3 Programs with Parallel Forms of Multiple Tests 270
10.3.1 Simultaneous Model 271
10.3.2 Item Overlap 272

10.3.3 Model with Aggregated Bounds 275
10.3.4 Discussion 276
10.4 Cost Function 276
10.4.1 Smoothing Cost Functions 277
10.5 Item Sets 278
10.5.1 Simultaneous Model 278
10.5.2 Three-Stage Approach 281
10.6 Calculating Solutions 282
10.7 Dynamic Versions of Design Models 284
10.7.1 Dynamic Models 284
10.7.2 Item Author as Attribute 286
10.7.3 Empirical Example 287
10.8 Assembling an Operational Item Pool 290
10.9 Final Comment 291
10.10 Literature 292
10.11 Summary 293
10.12 Exercises 294

11 Designing Item Pools for Programs with Adaptive Tests 297
11.1 Programs with a Single Adaptive Test 298
11.1.1 Design Model for Shadow Tests 298
11.1.2 Blueprint without Item-Exposure Control 301
11.1.3 Blueprint with Marginal Item-Exposure Control . 302
11.1.4 Blueprint with Conditional Item-Exposure Control 302
11.1.5 Empirical Example 303
11.2 Programs with Multiple Adaptive Tests 304
11.2.1 Different Tests from the Same Item Pool 304
11.2.2 Same Test from Different Item Pools 305
11.3 Item Sets 305
11.3.1 Design Model 305
11.3.2 Calculating the Blueprint 308
11.4 Calculating Shadow Tests 309
11.5 Some Remaining Topics 309
11.5.1 Stratifying an Item Pool 309
11.5.2 Empirical Example 310
11.5.3 Assembling an Item Pool as a Set of Fixed Test Forms 311
11.5.4 Empirical Example 313
11.5.5 Assembling a System of Rotating Item Pools . . . 314
11.5.6 Empirical Example 320
11.6 Literature 323
11.7 Summary 324
11.8 Exercises 325

12 Epilogue 327

Appendix 1: Basic Concepts in Linear Programming **333**
A1.1 Mathematical Programming 333
A1.1.1 Linear Programming 334
A1.1.2 Nonlinear Programming 335
A1.1.3 Other Forms of Mathematical Programming 335
A1.1.4 Constraints on Variables 336
A1.2 Graphical Example 337
A1.2.1 Problem 337
A1.2.2 Graphical Representation 338
A1.2.3 Number of Solutions 340
A1.3 Simplex Method 341
A1.4 Network-Flow Problems 342
A1.5 Solving Integer Problems 344
A1.6 Literature 346

Appendix 2: Example of a Test-Assembly Problem in *OPL Studio* **347**

Answers to Exercises **353**

Bibliography **389**

Index **403**