
Contents

| | |
|---|-------|
| Foreword | xiii |
| About This Book | xvii |
| Introduction | xxiii |
| Chapter 1. Operational Research | 1 |
| 1.1. A history | 1 |
| 1.2. Fields of application, principles and concepts | 2 |
| 1.2.1. Identification | 2 |
| 1.2.2. Modeling | 3 |
| 1.2.3. Solution | 5 |
| 1.2.4. Validation | 6 |
| 1.2.5. Implementation | 6 |
| 1.2.6. Improvement | 6 |
| 1.3. Basic models | 7 |
| 1.4. The future of OR | 7 |
| Chapter 2. Elements of Graph Theory | 9 |
| 2.1. Graphs and representations | 9 |
| 2.2. Undirected graph | 10 |
| 2.2.1. Multigraph | 10 |
| 2.2.2. Planar and non-planar graph | 11 |
| 2.2.3. Connected and unconnected graph | 11 |
| 2.2.4. Complete graph | 11 |
| 2.2.5. Bipartite graph | 12 |
| 2.2.6. Partial graph, subgraph, clique and stable | 12 |
| 2.2.7. Degree of a vertex and a graph | 13 |

| | |
|---|-----------|
| 2.2.8. Chain and cycle in a graph | 13 |
| 2.2.9. Level of connectivity (or beta index) | 15 |
| 2.2.10. Eulerian graph | 15 |
| 2.2.11. Hamiltonian graph | 16 |
| 2.2.12. Planar graph | 17 |
| 2.2.13. Isthmus | 18 |
| 2.2.14. Tree and forest | 19 |
| 2.2.15. Arborescence | 20 |
| 2.2.16. Ordered arborescence | 21 |
| 2.3. Directed graph or digraph | 22 |
| 2.3.1. Path and circuit in a digraph | 22 |
| 2.3.2. Absence of circuit in a digraph | 23 |
| 2.3.3. Adjacency matrix | 24 |
| 2.3.4. Valued graph matrix | 24 |
| 2.4. Graphs for logistics | 25 |
| Chapter 3. Optimal Paths | 27 |
| 3.1. Basic concepts | 27 |
| 3.2. Dijkstra's algorithm | 28 |
| 3.2.1. An example of calculating minimal paths | 28 |
| 3.2.2. Interpreting the results of the calculations | 30 |
| 3.3. Floyd–Warshall's algorithm | 30 |
| 3.3.1. Creating the starting matrices (initialization of the algorithm) | 30 |
| 3.3.2. Filling the matrices for the following repetitions | 31 |
| 3.3.3. An example of calculating minimal paths | 32 |
| 3.3.4. Interpreting the results | 34 |
| 3.4. Bellman–Ford's algorithm | 35 |
| 3.4.1. Initialization | 36 |
| 3.4.2. The next repetitions with relaxation | 36 |
| 3.4.3. An example of calculation | 37 |
| 3.4.4. Interpreting the results | 39 |
| 3.5. Bellman–Ford's algorithm with a negative circuit | 40 |
| 3.5.1. Example | 40 |
| 3.6. Exercises | 43 |
| 3.6.1. Exercise 1: Optimizing journey time | 43 |
| 3.6.2. Exercise 2: A directed graph with negative cost side | 44 |
| 3.6.3. Exercise 3: Routing data packets | 45 |
| 3.6.4. Solutions to exercise 1 | 45 |
| 3.6.5. Solutions to exercise 2 | 46 |
| 3.6.6. Solutions to exercise 3 | 48 |

| | |
|--|----|
| Chapter 4. Dynamic Programming | 51 |
| 4.1. The principles of dynamic programming | 51 |
| 4.2. Formulating the problem | 52 |
| 4.2.1. Example 1: The pyramid of numbers | 52 |
| 4.2.2. Example 2: The Fibonacci sequence. | 54 |
| 4.2.3. Example 3: The knapsack | 56 |
| 4.3. Stochastic process | 60 |
| 4.4. Markov chains | 60 |
| 4.4.1. Property of Markov chains | 61 |
| 4.4.2. Classes and states of a chain | 62 |
| 4.4.3. Matrix and graph | 63 |
| 4.4.4. Applying Markov chains | 64 |
| 4.5. Exercises | 66 |
| 4.5.1. Exercise 1: Levenshtein distance. | 66 |
| 4.5.2. Exercise 2 | 67 |
| 4.5.3. Exercise 3: Ehrenfest model | 67 |
| 4.5.4. Solutions to exercise 1 | 68 |
| 4.5.5. Solutions to exercise 2 | 69 |
| 4.5.6. Solutions to exercise 3 | 70 |
| | |
| Chapter 5. Scheduling with PERT and MPM | 73 |
| 5.1. Fundamental concepts | 73 |
| 5.2. Critical path method. | 74 |
| 5.3. Precedence diagram | 74 |
| 5.4. Planning a project with PERT-CPM | 77 |
| 5.4.1. A brief history. | 77 |
| 5.4.2. Methodology | 78 |
| 5.5. Example of determining a critical path with PERT | 86 |
| 5.5.1. Using the example to create a precedence table | 87 |
| 5.5.2. Creating the graph | 87 |
| 5.5.3. Numbering of vertices | 89 |
| 5.5.4. Determining earliest dates of each of the tasks | 89 |
| 5.5.5. Determining the latest dates for each of the tasks | 90 |
| 5.5.6. Determining the critical paths | 92 |
| 5.6. Slacks | 93 |
| 5.6.1. Total slack | 94 |
| 5.6.2. Free slack | 94 |
| 5.6.3. Certain slack (or independent slack). | 94 |
| 5.6.4. Properties | 94 |
| 5.7. Example of calculating slacks | 95 |
| 5.8. Determining the critical path with the help of a double-entry table | 96 |
| 5.8.1. Creating a table using our example | 96 |

| | |
|--|-----|
| 5.8.2 Filling out the table | 96 |
| 5.8.3. ES dates | 97 |
| 5.8.4. LF dates | 99 |
| 5.8.5. Critical path | 100 |
| 5.9. Methodology of planning with MPM | 101 |
| 5.9.1. A brief history | 101 |
| 5.9.2. Formalizing the graph | 102 |
| 5.9.3. Rules of construction. | 103 |
| 5.9.4. Earliest and latest dates | 104 |
| 5.9.5. Determining the critical path | 105 |
| 5.10. Example of determining a critical path with MPM | 106 |
| 5.10.1. Creating the graph. | 106 |
| 5.10.2. Determining the earliest dates for each task | 107 |
| 5.10.3. Determining the latest dates of each task | 108 |
| 5.10.4. Determining the critical path(s). | 109 |
| 5.10.5. Slacks | 109 |
| 5.11. Probabilistic PERT/CPM/MPM | 111 |
| 5.11.1. Probability of tasks | 112 |
| 5.11.2. Implementation in an example | 113 |
| 5.11.3. Calculating average durations and variance. | 114 |
| 5.11.4. Calculating the average duration of the project | 114 |
| 5.11.5. Calculating the probability of finishing the project in a chosen duration | 114 |
| 5.11.6. Calculating the duration of the project for a given probability | 115 |
| 5.12. Gantt diagram. | 116 |
| 5.12.1. Creating the diagram | 116 |
| 5.12.2. Example. | 117 |
| 5.13. PERT-MPM cost. | 119 |
| 5.13.1. Method | 120 |
| 5.13.2. Example. | 121 |
| 5.14. Exercises. | 125 |
| 5.14.1. Exercise 1. | 125 |
| 5.14.2. Exercise 2. | 126 |
| 5.14.3. Exercise 3. | 126 |
| 5.14.4. Exercise 4. | 127 |
| 5.14.5. Solutions to exercise 1 | 129 |
| 5.14.6. Solutions to exercise 2 | 130 |
| 5.14.7. Solutions to exercise 3 | 132 |
| 5.14.8. Solutions to exercise 4 | 133 |

| | |
|--|-----|
| Chapter 6. Maximum Flow in a Network | 137 |
| 6.1. Maximum flow | 137 |
| 6.2. Ford–Fulkerson algorithm | 138 |
| 6.2.1. Presentation of the algorithm | 139 |
| 6.2.2. Application of an example | 141 |
| 6.3. Minimum cut theorem | 147 |
| 6.3.1. Example of cuts | 148 |
| 6.4. Dinic algorithm | 149 |
| 6.4.1. Presenting the algorithm. | 149 |
| 6.4.2. Application in an example. | 150 |
| 6.5. Exercises | 154 |
| 6.5.1. Exercise 1: Drinking water supply | 154 |
| 6.5.2. Exercise 2: Maximum flow according to Dinic | 155 |
| 6.5.3. Solutions to exercise 1 | 155 |
| 6.5.4. Solutions to exercise 2 | 158 |
| | |
| Chapter 7. Trees, Tours and Transport | 163 |
| 7.1. The basic concepts. | 163 |
| 7.2. Kruskal’s algorithm | 165 |
| 7.2.1. Application to an example. | 165 |
| 7.3. Prim’s algorithm | 168 |
| 7.3.1. Application to an example. | 170 |
| 7.4. Sollin’s algorithm | 175 |
| 7.4.1. Application to an example. | 176 |
| 7.5. Little’s algorithm for solving the TSP | 182 |
| 7.5.1. Application to an example. | 184 |
| 7.6. Exercises | 195 |
| 7.6.1. Exercise 1: Computer network | 195 |
| 7.6.2. Exercise 2: Deliveries | 196 |
| 7.6.3. Solutions to exercise 1 | 197 |
| 7.6.4. Solutions to exercise 2 | 200 |
| | |
| Chapter 8. Linear Programming | 205 |
| 8.1. Basic concepts | 205 |
| 8.1.1. Formulation of a linear program | 206 |
| 8.2. The graphic resolution method | 206 |
| 8.2.1. Identification | 207 |
| 8.2.2. Formalization | 207 |
| 8.2.3. Resolution | 212 |
| 8.3. Simplex method | 215 |
| 8.3.1. Steps | 215 |

| | |
|--|------------|
| 8.3.2. An example to be addressed | 215 |
| 8.3.3. Formalization | 216 |
| 8.3.4. Change into standard form | 217 |
| 8.3.5. Creation of the table | 218 |
| 8.3.6. Determination of the pivot | 218 |
| 8.3.7. Iterations | 219 |
| 8.3.8. Interpretation | 222 |
| 8.4. Duality. | 223 |
| 8.4.1. Dual formulation | 224 |
| 8.4.2. Passage from primal to dual formalization | 224 |
| 8.4.3. Determination of the pivot | 226 |
| 8.4.4. Iterations | 227 |
| 8.4.5. Interpretation | 228 |
| 8.5. Exercises | 228 |
| 8.5.1. Exercise 1: Video and festival | 228 |
| 8.5.2. Exercise 2: Simplex | 228 |
| 8.5.3. Exercise 3: Primal and dual | 229 |
| 8.5.4. Solutions to exercise 1 | 229 |
| 8.5.5. Solutions to exercise 2 | 232 |
| 8.5.6. Solutions to exercise 3 | 234 |
| Chapter 9. Modeling Road Traffic | 237 |
| 9.1. A short introduction to road traffic | 237 |
| 9.2. Scale of models and networks | 239 |
| 9.3. Models and types | 239 |
| 9.4. Learning more information about the models. | 240 |
| 9.4.1. Microscopic models | 240 |
| 9.4.2. Macroscopic models | 245 |
| 9.4.3. The families of macroscopic models | 250 |
| 9.4.4. The discretization of models | 252 |
| 9.4.5. Mesoscopic models. | 253 |
| 9.4.6. Hybrid models. | 254 |
| 9.5. Urban modeling | 255 |
| 9.6. Intelligent transportation systems | 256 |
| 9.7. Conclusion | 256 |
| Chapter 10. Software Programs | 259 |
| 10.1. Software programs for OR and logistics | 259 |
| 10.2. Spreadsheets | 260 |
| 10.2.1. Existing software programs | 262 |

| | |
|---|-----|
| 10.3. Project managers | 266 |
| 10.3.1. The procedure for creating a project | 268 |
| 10.3.2. The different software programs available on the market | 268 |
| 10.4. Flow simulators | 271 |
| 10.4.1. Generalist software programs | 273 |
| 10.4.2. Pedestrian simulators | 281 |
| 10.4.3. Traffic simulators | 288 |
| 10.4.4. The creation of a simulation process | 300 |
| Appendices | 303 |
| Appendix 1: Standard Normal Distribution Table | 305 |
| Appendix 2: GeoGebra | 309 |
| Conclusion | 319 |
| Glossary | 323 |
| Bibliography | 329 |
| Index | 337 |