## Studies in Computational Intelligence, Volume 38

Editor-in-chief  
Prof. Janusz Kacprzyk  
Systems Research Institute  
Polish Academy of Sciences  
ul. Newelska 6  
01-447 Warsaw  
Poland  
E-mail: kacprzyk@ibspan.waw.pl  

Further volumes of this series can be found on our homepage: springer.com

<table>
<thead>
<tr>
<th>Volume</th>
<th>Editors</th>
<th>Title</th>
<th>ISBN</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vol. 22</td>
<td>N. Nedjah, E. Alba, L. de Macedo Mourelle</td>
<td>Parallel Evolutionary Computations</td>
<td>3-540-32837-8</td>
<td>2006</td>
</tr>
<tr>
<td>Vol. 24</td>
<td>Alakananda Bhattacharya, Amit Konar, Ajit K. Mandal</td>
<td>Parallel and Distributed Logic Programming</td>
<td>3-540-33458-0</td>
<td>2006</td>
</tr>
<tr>
<td>Vol. 25</td>
<td>Zoltán Ésik, Carlos Martín-Vide, Victor Mitrana</td>
<td>Recent Advances in Formal Languages and Applications</td>
<td>3-540-33460-2</td>
<td>2006</td>
</tr>
<tr>
<td>Vol. 26</td>
<td>Nadia Nedjah, Luiza de Macedo Mourelle</td>
<td>Swarm Intelligent Systems</td>
<td>3-540-33868-3</td>
<td>2006</td>
</tr>
<tr>
<td>Vol. 28</td>
<td>Brahim Chaib-draa, Jörg P. Möller</td>
<td>Multiagent based Supply Chain Management</td>
<td>3-540-33873-6</td>
<td>2006</td>
</tr>
<tr>
<td>Vol. 29</td>
<td>Sai Sumathi, S.N. Sivanandam</td>
<td>Introduction to Data Mining and its Applications</td>
<td>3-540-34689-9</td>
<td>2006</td>
</tr>
<tr>
<td>Vol. 30</td>
<td>Yukio Ohsawa, Shusaku Tsumoto</td>
<td>Chance Discoveries in Real World Decision Making</td>
<td>3-540-34352-0</td>
<td>2006</td>
</tr>
<tr>
<td>Vol. 31</td>
<td>Ajith Abraham, Crina Grosan, Vitorino Ramos</td>
<td>Stigmergic Optimization</td>
<td>3-540-34689-9</td>
<td>2006</td>
</tr>
<tr>
<td>Vol. 32</td>
<td>Akira Hirose</td>
<td>Complex-Valued Neural Networks</td>
<td>3-540-33456-4</td>
<td>2006</td>
</tr>
<tr>
<td>Vol. 33</td>
<td>Martin Pelikan, Kumara Sastry, Erick Cantú-Paz</td>
<td>Scalable Optimization via Probabilistic Modeling</td>
<td>3-540-34953-7</td>
<td>2006</td>
</tr>
<tr>
<td>Vol. 34</td>
<td>Ajith Abraham, Crina Grosan, Vitorino Ramos</td>
<td>Swarm Intelligence in Data Mining</td>
<td>3-540-34955-3</td>
<td>2006</td>
</tr>
<tr>
<td>Vol. 37</td>
<td>Jie Lu, Da Ruan, Guangquan Zhang</td>
<td>E-Service Intelligence</td>
<td>3-540-37015-3</td>
<td>2006</td>
</tr>
<tr>
<td>Vol. 38</td>
<td>Art Lew, Holger Mauch</td>
<td>Dynamic Programming</td>
<td>3-540-37013-7</td>
<td>2007</td>
</tr>
</tbody>
</table>
Dynamic Programming
A Computational Tool

With 55 Figures and 5 Tables
Prof. Art Lew  
Department of Information and Computer Sciences  
University of Hawaii at Manoa  
1680 East-West Road  
Honolulu, HI 96822  
USA  
E-mail: artlew@hawaii.edu

Dr. Holger Mauch  
Department of Computer Science  
Natural Sciences Collegium  
Eckerd College  
4200, 54th Ave. S.  
Saint Petersburg, FL 33711  
USA  
E-mail: mauchh@eckerd.edu

Library of Congress Control Number: 2006930743

ISSN print edition: 1860-949X  
ISSN electronic edition: 1860-9503  

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilm or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer-Verlag. Violations are liable to prosecution under the German Copyright Law.

Springer is a part of Springer Science+Business Media  
springer.com  
Springer-Verlag Berlin Heidelberg 2007

The use of general descriptive names, registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

Cover design: deblik, Berlin  
Typesetting by the authors and SPI  
Printed on acid-free paper  
SPIN: 11550860  
89/SPI  
5 4 3 2 1 0
To the Bellman Continuum, in memory of Richard Bellman. A.L.

To my family. H.M.
Dynamic programming has long been applied to numerous areas in mathematics, science, engineering, business, medicine, information systems, biomathematics, artificial intelligence, among others. Applications of dynamic programming have increased as recent advances have been made in areas such as neural networks, data mining, soft computing, and other areas of computational intelligence. The value of dynamic programming formulations and means to obtain their computational solutions has never been greater.

This book describes the use of dynamic programming as a computational tool to solve discrete optimization problems.

(1) We first formulate large classes of discrete optimization problems in dynamic programming terms, specifically by deriving the dynamic programming functional equations (DPFEs) that solve these problems. A text-based language, gDPS, for expressing these DPFEs is introduced. gDPS may be regarded as a high-level specification language, not a conventional procedural computer programming language, but which can be used to obtain numerical solutions.

(2) We then define and examine properties of Bellman nets, a class of Petri nets that serves both as a formal theoretical model of dynamic programming problems, and as an internal computer data structure representation of the DPFEs that solve these problems.

(3) We also describe the design, implementation, and use of a software tool, called DP2PN2Solver, for solving DPFEs. DP2PN2Solver may be regarded as a program generator, whose input is a DPFE, expressed in the input specification language gDPS and internally represented as a Bellman net, and whose output is its numerical solution that is produced indirectly by the generation of “solver” code, which when executed yields the desired solution.

This book should be of value to different classes of readers: students, instructors, practitioners, and researchers. We first provide a tutorial introduction to dynamic programming and to Petri nets. For those interested in dynamic programming, we provide a useful software tool that allows them to obtain numerical solutions. For researchers having an interest in the fields of
dynamic programming and Petri nets, unlike most past work which applies
dynamic programming to solve Petri net problems, we suggest ways to apply
Petri nets to solve dynamic programming problems.

For students and instructors of courses in which dynamic programming
is taught, usually as one of many other problem-solving methods, this book
provides a wealth of examples that show how discrete optimization problems
can be formulated in dynamic programming terms. Dynamic programming
has been and continues to be taught as an “art”, where how to use it must
be learned by example, there being no mechanical way to apply knowledge
of the general principles (e.g., the principle of optimality) to new unfamiliar
problems. Experience has shown that the greater the number and variety
of problems presented, the easier it is for students to apply general concepts.
Thus, one objective of this book is to include many and more diverse examples.
A further distinguishing feature of this book is that, for all of these examples,
we not only formulate the DP equations but also show their computational
solutions, exhibiting computer programs (in our specification language) as well
as providing as output numerical answers (as produced by the automatically
generated solver code).

In addition, we provide students and instructors with a software tool
(DP2PN2Solver) that enables them to obtain numerical solutions of dynamic
programming problems without requiring them to have much computer pro-
gramming knowledge and experience. This software tool can be downloaded
from either of the following websites:

http://natsci.eckerd.edu/~mauchh/Research/DP2PN2Solver
http://www2.hawaii.edu/~icl/DP2PN2Solver

Further information is given in Appendix B. Having such software support
allows them to focus on dynamic programming rather than on computer pro-
gramming. Since many problems can be solved by different dynamic program-
ning formulations, the availability of such a computational tool, that makes it
easier for readers to experiment with their own formulations, is a useful aid
to learning.

The DP2PN2Solver tool also enables practitioners to obtain numerical
solutions of dynamic programming problems of interest to them without
requiring them to write conventional computer programs. Their time, of
course, is better spent on problem formulation and analysis than on program
design and debugging. This tool allows them to verify that their formulations
are correct, and to revise them as may be necessary in their problem solving
efforts. The main limitation of this (and any) dynamic programming tool for
many practical problems is the size of the state space. Even in this event,
the tool may prove useful in the formulation stage to initially test ideas on
simplified scaled-down problems.

As a program generator, DP2PN2Solver is flexible, permitting alternate
front-ends and back-ends. Inputs other than in the gDPS language are possible.
Alternative DPFE specifications can be translated into gDPS or directly
into Bellman nets. Output solver code (i.e., the program that numerically solves a given DPFE) may be in alternative languages. The solver code emphasized in this book is Java code, largely because it is universally and freely available on practically every platform. We also discuss solver codes for spreadsheet systems and Petri net simulators. By default, the automatically generated solver code is hidden from the average user, but it can be inspected and modified directly by users if they wish.

Furthermore, this book describes research into connections between dynamic programming and Petri nets. It was our early research into such connections that ultimately lead to the concept of Bellman nets, upon which the development of our DP2PN2Solver tool is based. We explain here the underlying ideas associated with Bellman nets. Researchers interested in dynamic programming or Petri nets will find many open questions related to this work that suggest avenues of future research. For example, additional research might very likely result in improvements in the DP2PN2Solver tool, such as to address the state-space size issue or to increase its diagnostic capabilities. Every other aspect of this work may benefit from additional research.

Thus, we expect the DP2PN2Solver tool described in this book to undergo revisions from time to time. In fact, the tool was designed modularly to make it relatively easy to modify. As one example, changes to the gDPS specification language syntax can be made by simply revising its BNF definition since we use a compiler-compiler rather than a compiler to process it. Furthermore, alternate input languages (other than gDPS) and solver codes (other than Java) can be added as optional modules, without changing the existing modules. We welcome suggestions from readers on how the tool (or its description) can be improved. We may be contacted at artlew@hawaii.edu or mauchh@eckerd.edu. Updates to the software and to this book, including errata, will be placed on the aforementioned websites.

Acknowledgements. The authors wish to thank Janusz Kacprzyk for including this monograph in his fine series of books. His encouragement has been very much appreciated.

Honolulu, June 2006,  
Art Lew
St. Petersburg, June 2006,  
Holger Mauch
Contents

Part I Dynamic Programming

1 Introduction to Dynamic Programming .......................... 3
  1.1 Principles of Dynamic Programming ..................... 5
  1.1.1 Sequential Decision Processes ................. 6
  1.1.2 Dynamic Programming Functional Equations ....... 9
  1.1.3 The Elements of Dynamic Programming ........... 11
  1.1.4 Application: Linear Search ..................... 12
  1.1.5 Problem Formulation and Solution ............... 14
  1.1.6 State Transition Graph Model ................... 17
  1.1.7 Staged Decisions ............................. 19
  1.1.8 Path-States .................................. 21
  1.1.9 Relaxation .................................. 22
  1.1.10 Shortest Path Problems ....................... 23
  1.1.11 All-Pairs Shortest Paths ....................... 29
  1.1.12 State Space Generation ....................... 30
  1.1.13 Complexity .................................. 31
  1.1.14 Greedy Algorithms ........................... 32
  1.1.15 Probabilistic DP ............................. 32
  1.1.16 Nonoptimization Problems ..................... 33
  1.1.17 Concluding Remarks .......................... 34
  1.2 Computational Solution of DPFEs .................... 34
    1.2.1 Solution by Conventional Programming ........ 35
    1.2.2 The State-Decision-Reward-Transformation Table 36
    1.2.3 Code Generation ........................... 38
    1.2.4 Spreadsheet Solutions ....................... 38
    1.2.5 Example: SPA ............................. 40
    1.2.6 Concluding Remarks ......................... 42
  1.3 Overview of Book .................................. 42
# Applications of Dynamic Programming

2.1 Optimal Allotment Problem (ALLOT) .................................. 49
2.2 All-Pairs Shortest Paths Problem (APSP) .......................... 50
2.3 Optimal Alphabetic Radix-Code Tree Problem (ARC) ............ 51
2.4 Assembly Line Balancing (ASMBAL) ................................. 52
2.5 Optimal Assignment Problem (ASSIGN) ............................. 54
2.6 Optimal Binary Search Tree Problem (BST) ......................... 55
2.7 Optimal Covering Problem (COV) .................................... 57
2.8 Deadline Scheduling Problem (DEADLINE) ......................... 57
2.9 Discounted Profits Problem (DPP) .................................... 58
2.10 Edit Distance Problem (EDP) ........................................ 59
2.11 Fibonacci Recurrence Relation (FIB) ............................... 60
2.12 Flowshop Problem (FLOWSHOP) ................................... 61
2.13 Tower of Hanoi Problem (HANOI) .................................. 62
2.14 Integer Linear Programming (ILP) .................................. 63
2.15 Integer Knapsack as ILP Problem (ILPKNAP) ....................... 64
2.16 Interval Scheduling Problem (INTVL) ................................ 64
2.17 Inventory Problem (INVENT) .......................................... 66
2.18 Optimal Investment Problem (INVEST) ............................... 67
2.19 Investment: Winning in Las Vegas Problem (INVESTWL) ......... 68
2.20 0/1 Knapsack Problem (KS01) ...................................... 69
2.21 COV as KSINT Problem (KSCOV) .................................. 70
2.22 Integer Knapsack Problem (KSINT) .................................. 70
2.23 Longest Common Subsequence (LCS) ................................ 71
2.24 Optimal Linear Search Problem (LINSRC) ......................... 73
2.25 Lot Size Problem (LOT) ............................................... 73
2.26 Longest Simple Path Problem (LSP) ................................ 74
2.27 Matrix Chain Multiplication Problem (MCM) ....................... 75
2.28 Minimum Maximum Problem (MINMAX) ............................. 75
2.29 Minimum Weight Spanning Tree Problem (MWST) ................. 77
2.30 The Game of Nim (NIM) ............................................... 78
2.31 Optimal Distribution Problem (ODP) ............................... 80
2.32 Optimal Permutation Problem (PERM) ............................. 81
2.33 Jug-Pouring Problem (POUR) ....................................... 82
2.34 Optimal Production Problem (PROD) ............................... 83
2.35 Production: Reject Allowances Problem (PRODRAP) ............. 84
2.36 Reliability Design Problem (RDP) .................................. 84
2.37 Replacement Problem (REPLACE) ................................... 85
2.38 Stagecoach Problem (SCP) ........................................... 86
2.39 Seek Disk Scheduling Problem (SEEK) .............................. 87
2.40 Segmented Curve Fitting Problem (SEGLINE) ...................... 88
2.41 Program Segmentation Problem (SEGPAGE) ....................... 91
2.42 Optimal Selection Problem (SELECT) .............................. 94
2.43 Shortest Path in an Acyclic Graph (SPA) ......................... 95
2.44 Shortest Path in an Cyclic Graph (SPC) ............................. 95
### Part II Modeling of DP Problems

#### 3 The DP Specification Language gDPS

3.1 Introduction to gDPS .................................................. 103
3.2 Design Principles of gDPS ............................................. 105
3.3 Detailed Description of the gDPS Sections ......................... 106
  3.3.1 Name Section .................................................. 106
  3.3.2 General Variables Section .................................. 106
  3.3.3 Set Variables Section ......................................... 108
  3.3.4 General Functions Section ................................... 109
  3.3.5 State Type Section ............................................. 110
  3.3.6 Decision Variable Section .................................... 110
  3.3.7 Decision Space Section ........................................ 111
  3.3.8 Goal Section .................................................. 111
  3.3.9 DPFE Base Section ............................................. 112
  3.3.10 DPFE Section .................................................. 113
  3.3.11 Cost/Reward Function Section ............................... 115
  3.3.12 Transformation Function Section ............................ 115
  3.3.13 Transition Weight Section ................................... 116
3.4 BNF Grammar of the gDPS language .............................. 117

#### 4 DP Problem Specifications in gDPS

4.1 gDPS source for ALLOT ............................................. 125
4.2 gDPS source for APS .................................................. 128
4.3 gDPS source for ARC .................................................. 131
4.4 gDPS source for ASMBAL ............................................. 132
4.5 gDPS source for ASSIGN ............................................. 135
4.6 gDPS source for BST .................................................. 136
4.7 gDPS source for COV .................................................. 138
4.8 gDPS source for DEADLINE .......................................... 139
4.9 gDPS source for DPP .................................................. 140
4.10 gDPS source for EDP .................................................. 141
4.11 gDPS source for FIB .................................................. 144
4.12 gDPS source for FLOWSHOP ......................................... 144
4.13 gDPS source for HANOI ............................................. 145
4.14 gDPS source for ILP .................................................. 146
4.15 gDPS source for ILPKNAP ........................................... 148
4.16 gDPS source for INTVL .............................................. 150
4.17 gDPS source for INVENT ............................................ 154
4.18 gDPS source for INVEST ............................................ 156
5.3.4 The Role of Markings in the Low-Level Bellman Net Model ........................................ 214
5.3.5 Advantages of the Low-Level Bellman Net Model ......................................................... 214
5.4 Low-Level Bellman Net Properties .......................................................... 214
5.5 The High-Level Bellman Net Model ............................................................. 215
5.6 High-Level Bellman Net Properties ............................................................. 219

6 Bellman Net Representations of DP Problems ........................................ 221
   6.1 Graphical Representation of Low-Level Bellman Net Examples .............. 222
      6.1.1 Low-Level Bellman Net for BST ................................................................. 222
      6.1.2 Low-Level Bellman Net for LINSRC .............................................................. 222
      6.1.3 Low-Level Bellman Net for MCM ................................................................. 224
      6.1.4 Low-Level Bellman Net for ODP ................................................................. 224
      6.1.5 Low-Level Bellman Net for PERM ............................................................... 227
      6.1.6 Low-Level Bellman Net for SPA ................................................................. 228
   6.2 Graphical Representation of High-Level Bellman Net Examples .............. 228
      6.2.1 High-Level Bellman Net for EDP ................................................................. 230
      6.2.2 High-Level Bellman Net for ILP ................................................................. 230
      6.2.3 High-Level Bellman Net for KS01 ............................................................... 231
      6.2.4 High-Level Bellman Net for LCS ................................................................. 231
      6.2.5 High-Level Bellman Net for LINSRC ............................................................ 234
      6.2.6 High-Level Bellman Net for LSP ................................................................. 235
      6.2.7 High-Level Bellman Net for MCM ............................................................... 236
      6.2.8 High-Level Bellman Net for RDP ................................................................. 238
      6.2.9 High-Level Bellman Net for SCP ................................................................. 238
      6.2.10 High-Level Bellman Net for SPA ............................................................... 240
      6.2.11 High-Level Bellman Net for SPC ............................................................... 242

Part III Design and Implementation of DP Tool ........................................ 247

7 DP2PN2Solver Tool ............................................................. 247
   7.1 Overview ................................................................. 247
   7.2 Internal Representation of Bellman Nets ....................................................... 251
   7.3 Compiling and Executing DP Programs ......................................................... 252
   7.4 The ILP2gDPS Preprocessor Module ............................................................ 255

8 DP2PN Parser and Builder ............................................................. 259
   8.1 Design of the DP2PN modules ................................................................. 259
   8.2 Implementation of the DP2PN modules ....................................................... 260
   8.3 The Module LINSRCMain ................................................................. 263
   8.4 Error Detection in DP2PN ................................................................. 268
# 9 The PN2Solver Modules

9.1 The Solver Code Generation Process .................................. 271
9.2 The PN2Java Module .................................................. 273
  9.2.1 Java Solver Code Calculation Objects .......................... 274
  9.2.2 Java Solver Code for LINSRCS ................................. 276
  9.2.3 Java Solver Code for LSP ...................................... 278
  9.2.4 Java Solver Code for MCM .................................... 278
  9.2.5 Java Solver Code for SPA .................................... 280
9.3 The PN2Spreadsheet Module ......................................... 281
  9.3.1 PN2Spreadsheet Solver Code for LINSRCS ...................... 282
  9.3.2 PN2Spreadsheet Solver Code for Other Examples .............. 284
9.4 The PN2XML Module .................................................. 284
  9.4.1 Petri Net Solver Code for LINSRCS ........................... 285
  9.4.2 Petri Net Solver Code for SPA ................................ 288
9.5 Conclusion .................................................................. 289

# Part IV Computational Results

10 Java Solver Results of DP Problems ................................. 293
  10.1 ALLOT Java Solver Output ....................................... 293
  10.2 APSP Java Solver Output .......................................... 294
  10.3 ARC Java Solver Output ............................................ 296
  10.4 ASMBAL Java Solver Output ...................................... 296
  10.5 ASSIGN Java Solver Output ....................................... 297
  10.6 BST Java Solver Output ............................................ 297
  10.7 COV Java Solver Output ............................................ 298
  10.8 DEADLINE Java Solver Output ................................... 298
  10.9 DPP Java Solver Output ............................................ 299
  10.10 EDP Java Solver Output ......................................... 299
  10.11 FIB Java Solver Output .......................................... 299
  10.12 FLOWSHOP Java Solver Output ................................. 300
  10.13 HANOI Java Solver Output ...................................... 300
  10.14 ILP Java Solver Output .......................................... 301
  10.15 ILPKNAP Java Solver Output .................................... 301
  10.16 INTVL Java Solver Output ....................................... 302
  10.17 INVENT Java Solver Output ..................................... 303
  10.18 INVEST Java Solver Output ..................................... 304
  10.19 INVESTWLV Java Solver Output ............................... 304
  10.20 KS01 Java Solver Output ........................................ 305
  10.21 KSCOV Java Solver Output .................................... 306
  10.22 KSINT Java Solver Output ..................................... 306
  10.23 LCS Java Solver Output .......................................... 306
  10.24 LINSRC Java Solver Output .................................... 307
  10.25 LOT Java Solver Output ......................................... 308
10.26 LSP Java Solver Output ........................................... 308
10.27 MCM Java Solver Output ........................................... 308
10.28 MINMAX Java Solver Output ........................................ 309
10.29 MWST Java Solver Output ........................................... 309
10.30 NIM Java Solver Output ............................................ 309
10.31 ODP Java Solver Output ............................................ 312
10.32 PERM Java Solver Output ........................................... 312
10.33 POUR Java Solver Output ........................................... 312
10.34 PROD Java Solver Output ........................................... 313
10.35 PRODRAP Java Solver Output ........................................ 314
10.36 RDP Java Solver Output ............................................ 314
10.37 REPLACE Java Solver Output ........................................ 315
10.38 SCP Java Solver Output ............................................. 315
10.39 SEEK Java Solver Output ............................................ 315
10.40 SEGLINE Java Solver Output ........................................ 316
10.41 SEGPAGE Java Solver Output ....................................... 316
10.42 SELECT Java Solver Output .......................................... 317
10.43 SPA Java Solver Output ............................................. 317
10.44 SPC Java Solver Output ............................................. 318
10.45 SPT Java Solver Output ............................................. 318
10.46 TRANSPo Java Solver Output ......................................... 319
10.47 TSP Java Solver Output ............................................. 319

11 Other Solver Results .................................................. 321
11.1 PN2Spreadsheet Solver Code Output .................................. 321
11.1.1 PN2Spreadsheet Solver Code for LINSRCS ....................... 321
11.1.2 PN2Spreadsheet Solver Code for LSP ............................ 322
11.1.3 PN2Spreadsheet Solver Code for MCM ............................ 322
11.1.4 PN2Spreadsheet Solver Code for SPA ............................ 323
11.1.5 Spreadsheet Output ................................................ 323
11.2 PN2XML Solver Code Output .......................................... 324
11.2.1 PN2XML Simulation Output for LINSRCS ....................... 325

12 Conclusions ............................................................ 329
12.1 Applicability of DP and DP2PN2Solver .............................. 329
12.2 The DP2PN2Solver Tool .............................................. 330
12.3 Research Directions ................................................... 332
12.3.1 User Functionality ................................................. 333
12.3.2 Reduction of Dimensionality ..................................... 334
12.3.3 Petri Net Modeling ................................................. 335
12.4 Summary .............................................................. 336
A Supplementary Material ............................................. 339
   A.1 Pseudocode of the DP2PN Module .............................. 339
      A.1.1 Main Class for LINSRCS ............................... 339
      A.1.2 State Class for LINSRCS ............................. 342
      A.1.3 Decision Class .............................. 343
      A.1.4 DPInstanceTableEntry Class ......................... 344
      A.1.5 DPInstance Class .................................. 344
      A.1.6 BellmanNet Class .................................... 349
   A.2 DP2PN System Files ........................................ 353
   A.3 Output from PN2XML ........................................ 356
      A.3.1 High-Level Bellman Net XML file for SPA1 ............. 356
B User Guide for DP2PN2Solver ...................................... 359
   B.1 System Requirements for DP2PN2Solver ....................... 359
      B.1.1 Java Environment .................................. 359
   B.2 Obtaining DP2PN2Solver .................................. 360
   B.3 Installation of DP2PN2Solver ............................... 360
      B.3.1 Deployment of the Files ......................... 360
   B.4 Running DP2PN2Solver .................................... 361
      B.4.1 The DP2PN Module ................................. 361
      B.4.2 The PN2Solver Module ............................ 363
   B.5 Creation of the gDPS Source File ........................... 365
   B.6 Debugging gDPS Code ..................................... 366
      B.6.1 Omission of Base Cases ............................. 366
      B.6.2 Common Mistakes .............................. 367
   B.7 Error Messages of DP2PN2Solver ............................ 368

References .......................................................... 371

Index ............................................................... 375