

Hans Kellerer, Ulrich Pferschy, David Pisinger

Knapsack Problems

Springer's internal project number, if known

Mathematics – Monograph (English)

October 10, 2003

Springer-Verlag

Berlin Heidelberg New York

London Paris Tokyo

Hong Kong Barcelona

Budapest

Preface

Thirteen years have passed since the seminal book on knapsack problems by Martello and Toth appeared. On this occasion a former colleague exclaimed back in 1990: “How can you write 250 pages on the knapsack problem?” Indeed, the definition of the knapsack problem is easily understood even by a non-expert who will not suspect the presence of challenging research topics in this area at the first glance.

However, in the last decade a large number of research publications contributed new results for the knapsack problem in all areas of interest such as exact algorithms, heuristics and approximation schemes. Moreover, the extension of the knapsack problem to higher dimensions both in the number of constraints and in the number of knapsacks, as well as the modification of the problem structure concerning the available item set and the objective function, leads to a number of interesting variations of practical relevance which were the subject of intensive research during the last few years.

Hence, two years ago the idea arose to produce a new monograph covering not only the most recent developments of the standard knapsack problem, but also giving a comprehensive treatment of the whole knapsack family including the siblings such as the subset sum problem and the bounded and unbounded knapsack problem, and also more distant relatives such as multidimensional, multiple, multiple-choice and quadratic knapsack problems in dedicated chapters.

Furthermore, attention is paid to a number of less frequently considered variants of the knapsack problem and to the study of stochastic aspects of the problem. To illustrate the high practical relevance of the knapsack family for many industrial and economic problems, a number of applications are described in more detail. They are selected subjectively from the innumerable occurrences of knapsack problems reported in the literature.

Our above-mentioned colleague will be surprised to notice that even on the more than 500 pages of this book not all relevant topics could be treated in equal depth but decisions had to be made on where to go into details of constructions and proofs and where to concentrate on stating results and refer to the appropriate publications. Moreover, an editorial deadline had to be drawn at some point. In our case, we stopped looking for new publications at the end of June 2003.

The audience we envision for this book is threefold: The first two chapters offer a very basic introduction to the knapsack problem and the main algorithmic concepts to derive optimal and approximate solution. Chapter 3 presents a number of advanced algorithmic techniques which are used throughout the later chapters of the book. The style of presentation in these three chapters is kept rather simple and assumes only minimal prerequisites. They should be accessible to students and graduates of business administration, economics and engineering as well as practitioners with little knowledge of algorithms and optimization.

This first part of the book is also well suited to introduce classical concepts of optimization in a classroom, since the knapsack problem is easy to understand and is probably the least difficult but most illustrative problem where dynamic programming, branch-and-bound, relaxations and approximation schemes can be applied.

In these chapters no knowledge of linear or integer programming and only a minimal familiarity with basic elements of graph theory is assumed. The issue of \mathcal{NP} -completeness is dealt with by an intuitive introduction in Section 1.5, whereas a thorough and rigorous treatment is deferred to the Appendix.

The remaining chapters of the book address two different audiences. On one hand, a student or graduate of mathematics or computer science, or a successful reader of the first three chapters willing to go into more depth, can use this book to study advanced algorithms for the knapsack problem and its relatives. On the other hand, we hope scientific researchers or expert practitioners will find the book a valuable source of reference for a quick update on the state of the art and on the most efficient algorithms currently available. In particular, a collection of computational experiments, many of them published for the first time in this book, should serve as a valuable tool to pick the algorithm best suited for a given problem instance. To facilitate the use of the book as a reference we tried to keep these chapters self-contained as far as possible.

For these advanced audiences we assume familiarity with the basic theory of linear programming, elementary elements of graph theory, and concepts of algorithms and data structures as far as they are generally taught in basic courses on these subjects.

Chapters 4 to 12 give detailed presentations of the knapsack problem and its variants in increasing order of structural difficulty. Hence, we start with the subset sum problem in Chapter 4, move on to the standard knapsack problem which is discussed extensively in two chapters, one for exact and one for approximate algorithms, and finish this second part of the book with the bounded and unbounded knapsack problem in Chapters 7 and 8.

The third part of the book contains more complicated generalizations of the knapsack problems. It starts with the multidimensional knapsack problem (a knapsack problem with d constraints) in Chapter 9, then considers the multiple knapsack problem (m knapsacks are available for packing) in Chapter 10, goes on to the multiple-choice knapsack problem (the items are partitioned into classes and exactly one item of each class must be packed), and extends the linear objective func-

tion to a quadratic one yielding the quadratic knapsack problem in Chapter 12. This chapter also contains an excursion to semidefinite programming giving a mostly self-contained short introduction to this topic.

A collection of other variants of the knapsack problem is put together in Chapter 13. Detailed expositions are devoted to the multiobjective and the precedence constraint knapsack problem, whereas other subjectively selected variants are treated in a more cursory way. The solitary Chapter 14 gives a survey on stochastic results for the knapsack problem. It also contains a section on the on-line version of the problem.

All these six chapters can be seen as survey articles, most of them being the first survey on their subject, containing many pointers to the literature and some examples of application.

Particular effort was put into the description of interesting applications of knapsack type problems. We decided to avoid a boring listing of umpteen papers with a two-line description of the occurrence of a knapsack problem for each of them, but selected a smaller number of application areas where knapsack models play a prominent role. These areas are discussed in more detail in Chapter 15 to give the reader a full understanding of the situations presented. They should be particularly useful for teaching purposes.

The Appendix gives a short presentation of \mathcal{NP} -completeness with the focus on knapsack problems. Without venturing into the depths of theoretical computer science and avoiding topics such as Turing machines and unary encoding, a rather informal introduction to \mathcal{NP} -completeness is given, however with formal proofs for the \mathcal{NP} -hardness of the subset sum and the knapsack problem.

Some assumptions and conventions concerning notation and style are kept throughout the book. Most algorithms are stated in a flexible pseudocode style putting emphasis on readability instead of formal uniformity. This means that simpler algorithms are given in the style of an unknown but easily understandable programming language, whereas more complex algorithms are introduced by a structured, but verbal description. Commands and verbal instructions are given in **Sans Serif** font, whereas comments follow in *Italic* letters. As a general reference and guideline to algorithms we used the book by Cormen, Leiserson, Rivest and Stein [92].

For the sake of readability and personal taste we follow the non-standard convention of using the term *increasing* instead of the mathematically correct *nondecreasing* and in the same way *decreasing* instead of *nonincreasing*. Wherever we use the log function we always refer to the base 2 logarithm unless stated otherwise. After the Preface we give a short list of notations containing only those terms which are used throughout the book. Many more naming conventions will be introduced on a local level during the individual chapters and sections.

As mentioned above a number of computational experiments were performed for exact algorithms. These were performed on the following machines:

AMD ATHLON, 1.2 GHZ	SPECint2000 = 496	SPECfp2000 = 417
INTEL PENTIUM 4, 1.5 GHZ	SPECint2000 = 558	SPECfp2000 = 615
INTEL PENTIUM III, 933 MHZ	SPECint2000 = 403	SPECfp2000 = 328

The performance index was obtained from SPEC (www.specbench.org). As can be seen the three machines have reasonably similar performance, making it possible to compare running times across chapters. The codes have been compiled using the GNU project C and C++ Compiler gcc-2.96, which also compiles Fortran77 code, thus preventing differences in computation times due to alternative compilers.

Acknowledgements

The authors strongly believe in the necessity to do research not with an island mentality but in an open exchange of knowledge, opinions and ideas within an international research community. Clearly, none of us would have been able to contribute to this book without the innumerable personal exchanges with colleagues on conferences and workshops, in person, by e-mail or even by surface mail. Therefore, we would like to start our acknowledgements by thanking the global research community for providing the spirit necessary for joint projects of collection and presentation.

The classic book by Silvano Martello and Paolo Toth on knapsack problems was frequently used as a reference during the writing of this text. Comments by both authors were greatly appreciated.

To our personal friends Alberto Caprara and Eranda Cela we owe special thanks for many discussions and helpful suggestions. John M. Bergstrom brought to our attention the importance of solving knapsack problems for all values of the capacity. Jarl Friis gave valuable comments on the chapter on the quadratic knapsack problem. Klaus Ladner gave valuable technical support, in particular in the preparation of figures.

In the computational experiments and comparisons, codes were used which were made available by Martin E. Dyer, Silvano Martello, Nei Y. Soma, Paolo Toth and John Walker. We thank them for their cooperation. Anders Bo Rasmussen and Rune Sandvik deserve special thanks for having implemented the upper bounds for the quadratic knapsack problem in Chapter 12 and for having run the computational experiments with these bounds. In this context the authors would also like to acknowledge DIKU Copenhagen for having provided the computational facilities for the computational experiments.

Finally, we would like to thank the Austrian and Danish tax payer for enabling us to devote most of our concentration on the writing of this book during the last two years. We would also like to apologize to the colleagues of our departments, our friends and our families for having neglected them during this time. Further apologies go to the reader of this book for any errors and mistakes it contains. These will be collected at the web-site of this book at www.diku.dk/knapsack.

Table of Contents

Preface	V
Table of Contents	IX
List of Notations	XIX
1. Introduction	1
1.1 Introducing the Knapsack Problem	1
1.2 Variants and Extensions of the Knapsack Problem	5
1.3 Single-Capacity Versus All-Capacities Problem	9
1.4 Assumptions on the Input Data	9
1.5 Performance of Algorithms	11
2. Basic Algorithmic Concepts	15
2.1 The Greedy Algorithm	15
2.2 Linear Programming Relaxation	17
2.3 Dynamic Programming	20
2.4 Branch-and-Bound	27
2.5 Approximation Algorithms	29
2.6 Approximation Schemes	37

3.	Advanced Algorithmic Concepts	43
3.1	Finding the Split Item in Linear Time	43
3.2	Variable Reduction	44
3.3	Storage Reduction in Dynamic Programming	46
3.4	Dynamic Programming with Lists	50
3.5	Combining Dynamic Programming and Upper Bounds	53
3.6	Balancing	54
3.7	Word RAM Algorithms	60
3.8	Relaxations	62
3.9	Lagrangian Decomposition	65
3.10	The Knapsack Polytope	67
4.	The Subset Sum Problem	73
4.1	Dynamic Programming	75
4.1.1	Word RAM Algorithm	76
4.1.2	Primal-Dual Dynamic Programming Algorithms	79
4.1.3	Primal-Dual Word-RAM Algorithm	80
4.1.4	Horowitz and Sahni Decomposition	81
4.1.5	Balancing	82
4.1.6	Bellman Recursion in Decision Form	85
4.2	Branch-and-Bound	85
4.2.1	Upper Bounds	86
4.2.2	Hybrid Algorithms	87
4.3	Core Algorithms	88
4.3.1	Fixed Size Core	89
4.3.2	Expanding Core	89
4.3.3	Fixed Size Core and Decomposition	90
4.4	Computational Results: Exact Algorithms	90
4.4.1	Solution of All-Capacities Problems	93
4.5	Polynomial Time Approximation Schemes for Subset Sum	94
4.6	A Fully Polynomial Time Approximation Scheme for Subset Sum	97
4.7	Computational Results: FPTAS	112

5. Exact Solution of the Knapsack Problem	117
5.1 Branch-and-Bound	119
5.1.1 Upper Bounds for (KP)	119
5.1.2 Lower Bounds for (KP)	124
5.1.3 Variable Reduction	125
5.1.4 Branch-and-Bound Implementations	127
5.2 Primal Dynamic Programming Algorithms	130
5.2.1 Word RAM Algorithm	131
5.2.2 Horowitz and Sahni Decomposition	136
5.3 Primal-Dual Dynamic Programming Algorithms	136
5.3.1 Balanced Dynamic Programming	138
5.4 The Core Concept	140
5.4.1 Finding a Core	142
5.4.2 Core Algorithms	144
5.4.3 Combining Dynamic Programming with Tight Bounds ...	147
5.5 Computational Experiments	150
5.5.1 Difficult Instances	154
5.5.2 Difficult Instances with Large Coefficients	155
5.5.3 Difficult Instances With Small Coefficients	156
6. Approximation Algorithms for the Knapsack Problem	161
6.1 Polynomial Time Approximation Schemes	161
6.1.1 Improving the <i>PTAS</i> for (KP)	161
6.2 Fully Polynomial Time Approximation Schemes	166
6.2.1 Scaling and Reduction of the Item Set	169
6.2.2 An Auxiliary Vector Merging Problem	171
6.2.3 Solving the Reduced Problem	175
6.2.4 Putting the Pieces Together	177

7. The Bounded Knapsack Problem	185
7.1 Introduction	185
7.1.1 Transformation of (BKP) into (KP)	187
7.2 Dynamic Programming	190
7.2.1 A Minimal Algorithm for (BKP)	191
7.2.2 Improved Dynamic Programming: Reaching (KP) Complexity for (BKP)	194
7.2.3 Word RAM Algorithm	200
7.2.4 Balancing	200
7.3 Branch-and-Bound	201
7.3.1 Upper Bounds	201
7.3.2 Branch-and-Bound Algorithms	202
7.3.3 Computational Experiments	204
7.4 Approximation Algorithms	205
8. The Unbounded Knapsack Problem	211
8.1 Introduction	211
8.2 Periodicity and Dominance	214
8.2.1 Periodicity	215
8.2.2 Dominance	216
8.3 Dynamic Programming	219
8.3.1 Some Basic Algorithms	220
8.3.2 An Advanced Algorithm	223
8.3.3 Word RAM Algorithm	227
8.4 Branch-and-Bound	228
8.5 Approximation Algorithms	232
9. Multidimensional Knapsack Problems	235
9.1 Introduction	235
9.2 Relaxations and Reductions	238
9.3 Exact Algorithms	246
9.3.1 Branch-and-Bound Algorithms	246

9.3.2	Dynamic Programming	248
9.4	Approximation	252
9.4.1	Negative Approximation Results	252
9.4.2	Polynomial Time Approximation Schemes	254
9.5	Heuristic Algorithms	255
9.5.1	Greedy-Type Heuristics	256
9.5.2	Relaxation-Based Heuristics	261
9.5.3	Advanced Heuristics	264
9.5.4	Approximate Dynamic Programming	266
9.5.5	Metaheuristics	268
9.6	The Two-Dimensional Knapsack Problem	269
9.7	The Cardinality Constrained Knapsack Problem	271
9.7.1	Related Problems	272
9.7.2	Branch-and-Bound	273
9.7.3	Dynamic Programming	273
9.7.4	Approximation Algorithms	276
9.8	The Multidimensional Multiple-Choice Knapsack Problem	280
10.	Multiple Knapsack Problems	285
10.1	Introduction	285
10.2	Upper Bounds	288
10.2.1	Variable Reduction and Tightening of Constraints	291
10.3	Branch-and-Bound	292
10.3.1	The MTM Algorithm	293
10.3.2	The Mulknep Algorithm	294
10.3.3	Computational Results	296
10.4	Approximation Algorithms	298
10.4.1	Greedy-Type Algorithms and Further Approximation Algorithms	299
10.4.2	Approximability Results for (B-MSSP)	301
10.5	Polynomial Time Approximation Schemes	304
10.5.1	A PTAS for the Multiple Subset Problem	304

10.5.2	A <i>PTAS</i> for the Multiple Knapsack Problem	311
10.6	Variants of the Multiple Knapsack Problem	315
10.6.1	The Multiple Knapsack Problem with Assignment Restrictions	315
10.6.2	The Class-Constrained Multiple Knapsack Problem	315
11.	The Multiple-Choice Knapsack Problem	317
11.1	Introduction	317
11.2	Dominance and Upper Bounds	319
11.2.1	Linear Time Algorithms for the LP-Relaxed Problem	322
11.2.2	Bounds from Lagrangian Relaxation	325
11.2.3	Other Bounds	327
11.3	Class Reduction	327
11.4	Branch-and-Bound	328
11.5	Dynamic Programming	329
11.6	Reduction of States	331
11.7	Hybrid Algorithms and Expanding Core Algorithms	332
11.8	Computational Experiments	335
11.9	Heuristics and Approximation Algorithms	338
11.10	Variants of the Multiple-Choice Knapsack Problem	339
11.10.1	Multiple-Choice Subset Sum Problem	339
11.10.2	Generalized Multiple-Choice Knapsack Problem	340
11.10.3	The Knapsack Sharing Problem	342
12.	The Quadratic Knapsack Problem	349
12.1	Introduction	349
12.2	Upper Bounds	351
12.2.1	Continuous Relaxation	352
12.2.2	Bounds from Lagrangian Relaxation of the Capacity Constraint	352
12.2.3	Bounds from Upper Planes	355
12.2.4	Bounds from Linearisation	356

12.2.5	Bounds from Reformulation	359
12.2.6	Bounds from Lagrangian Decomposition	362
12.2.7	Bounds from Semidefinite Relaxation	367
12.3	Variable Reduction	373
12.4	Branch-and-Bound	374
12.5	The Algorithm by Caprara, Pisinger and Toth	375
12.6	Heuristics	379
12.7	Approximation Algorithms	380
12.8	Computational Experiments — Exact Algorithms	382
12.9	Computational Experiments — Upper Bounds	384
13.	Other Knapsack Problems	389
13.1	Multiobjective Knapsack Problems	389
13.1.1	Introduction	389
13.1.2	Exact Algorithms for (MOKP)	391
13.1.3	Approximation of the Multiobjective Knapsack Problem	393
13.1.4	An <i>FPTAS</i> for the Multiobjective Knapsack Problem	395
13.1.5	A <i>PTAS</i> for (M ₀ d-KP)	397
13.1.6	Metaheuristics	401
13.2	The Precedence Constraint Knapsack Problem (PCKP)	402
13.2.1	Dynamic Programming Algorithms for Trees	404
13.2.2	Other Results for (PCKP)	407
13.3	Further Variants	408
13.3.1	Nonlinear Knapsack Problems	409
13.3.2	The Max-Min Knapsack Problem	411
13.3.3	The Minimization Knapsack Problem	412
13.3.4	The Equality Knapsack Problem	413
13.3.5	The Strongly Correlated Knapsack Problem	414
13.3.6	The Change-Making Problem	415
13.3.7	The Collapsing Knapsack Problem	416
13.3.8	The Parametric Knapsack Problem	419
13.3.9	The Fractional Knapsack Problem	421

13.3.10	The Set-Union Knapsack Problem	423
13.3.11	The Multiperiod Knapsack Problem	424
14.	Stochastic Aspects of Knapsack Problems	425
14.1	The Probabilistic Model	426
14.2	Structural Results	427
14.3	Algorithms with Expected Performance Guarantee	430
14.3.1	Related Models and Algorithms	431
14.4	Expected Performance of Greedy-Type Algorithms	433
14.5	Algorithms with Expected Running Time	436
14.6	Results for the Subset Sum Problem	437
14.7	Results for the Multidimensional Knapsack Problem	440
14.8	The On-Line Knapsack Problem	442
14.8.1	Time Dependent On-Line Knapsack Problems	445
15.	Some Selected Applications	449
15.1	Two-Dimensional Two-Stage Cutting Problems	449
15.1.1	Cutting a Given Demand from a Minimal Number of Sheets	450
15.1.2	Optimal Utilization of a Single Sheet	452
15.2	Column Generation in Cutting Stock Problems	455
15.3	Separation of Cover Inequalities	459
15.4	Financial Decision Problems	461
15.4.1	Capital Budgeting	461
15.4.2	Portfolio Selection	462
15.4.3	Interbank Clearing Systems	464
15.5	Asset-Backed Securitization	465
15.5.1	Introducing Securitization and Amortization Variants	466
15.5.2	Formal Problem Definition	468
15.5.3	Approximation Algorithms	469
15.6	Knapsack Cryptosystems	472
15.6.1	The Merkle-Hellman Cryptosystem	473

15.6.2	Breaking the Merkle-Hellman Cryptosystem	475
15.6.3	Further Results on Knapsack Cryptosystems	477
15.7	Combinatorial Auctions	478
15.7.1	Multi-Unit Combinatorial Auctions and Multi-Dimensional Knapsacks	479
15.7.2	A Multi-Unit Combinatorial Auction Problem with Decreasing Costs per Unit	481
A.	Introduction to \mathcal{NP}-Completeness of Knapsack Problems	483
A.1	Definitions	483
A.2	\mathcal{NP} -Completeness of the Subset Sum Problem	487
A.2.1	Merging of Constraints	488
A.2.2	\mathcal{NP} -Completeness	490
A.3	\mathcal{NP} -Completeness of the Knapsack Problem	491
A.4	\mathcal{NP} -Completeness of Other Knapsack Problems	491
	References	495
	Author Index	527
	Subject Index	535