Available online at www.sciencedirect.com
SCIENCE DIRECT®
ELSEVIER
European Journal of Operational Research 155 (2004) 1–21
EUROPEAN JOURNAL OF OPERATIONAL RESEARCH
www.elsevier.com/locate/dsw

Invited Review

The multidimensional 0–1 knapsack problem: An overview

Arnaud Fréville *

Laboratoire d'Automatique, de Mécanique et d'Informatique, Industrielles et Humaines, UMR CNRS 8530, Groupe Recherche Opérationnelle et Informatique, Université de Valenciennes et du Hainaut-Cambrésis, Le Mont Houy, 59313 Valenciennes Cedex, France

Received 18 January 2002; accepted 15 January 2003

Abstract

The multidimensional 0–1 knapsack problem is one of the most well-known integer programming problems and has received wide attention from the operational research community during the last four decades. Although recent advances have made possible the solution of medium size instances, solving this NP-hard problem remains a very interesting challenge, especially when the number of constraints increases. This paper surveys the main results published in the literature. The focus is on the theoretical properties as well as approximate or exact solutions of this special 0–1 program.

Keywords: Multidimensional 0–1 knapsack problem; Heuristics; Probabilistic and worst-case analysis; Surrogate duality; Preprocessing; Branch-and-bound algorithms

1. Introduction

The multidimensional 0–1 knapsack problem (MKP) is a special case of general linear 0–1 programs. Several names have been mentioned in the literature for the MKP: m-dimensional knapsack problem, multidimensional knapsack problem, multiknapsack problem, multiconstraint 0–1 knapsack problem, etc... We choose to refer to the name coined first by Weingartner and Ness [184], which means without ambiguity that MKP is a generalization of the standard 0–1 knapsack problem ($m = 1$).

Historically, the first examples have been exhibited by Lorie and Savage [120] and by Manne and Markowitz [126] as a capital budgeting model. Basically, the MKP is a resource allocation model which can be stated as

$$\max \quad z = \sum_{j=1}^{n} c_j x_j$$

$$\text{s.t.} \quad \sum_{j=1}^{n} a_{ij} x_j \leqslant b_i, \quad i \in M = \{1, 2, \ldots, m\}, \tag{1}$$

$$x_j \in \{0, 1\}, \quad j \in N = \{1, 2, \ldots, n\}, \tag{2}$$

where n is the number of items and m is the number of knapsack constraints with capacities b_i

* Address: Directeur des Transports, Conseil Regional Nord-Pas de Calais, 45 D, rue de Tournai, 59555 Lille Cedex, France.
E-mail addresses: arnaud.freville@univ-valenciennes.fr, a.freville@cr-npdc.fr (A. Fréville).

0377-2217/$ - see front matter © 2003 Elsevier B.V. All rights reserved.
doi:10.1016/S0377-2217(03)00274-1

$(i = 1, 2, \ldots, m)$. Each item $j \in N$ requires a_{ij} units of resource consumption in the ith knapsack $(i = 1, 2, \ldots, m)$ and yields c_j units of profit upon inclusion. The goal is to find a subset of items that yields maximum profit without exceeding the resource capacities. By its nature, all entries are nonnegative. More precisely, it can be assumed, without loss of generality, that $c_j > 0$, $b_i > 0$, $0 \leqslant a_{ij} \leqslant b_i$ and $\sum_{j=1}^{n} a_{ij} > b_i$ for all $j \in N$ and for all $i \in M$ (since otherwise some or all of the variables can fixed to 0 or 1). Moreover, any MKP having at least one of these entries a_{ij} equal to 0, can be replaced by an equivalent MKP with positive entries, i.e. both problems have the same feasible solutions.

The MKP is one of the most well-known constrained integer programming problem, and the large domain of its applications has greatly contributed to its fame. Following the seminal paper of Lorie and Savage [120], MKP modeling has been investigated intensively in capital budgeting and project selection applications [135,146,183]. Recently, Meier et al. [137] investigated more realistic approaches combining capital budgeting models with novel, real techniques for project evaluation. One of their contributions was to propose a new scenario-based capital budgeting model which includes the MKP as a subproblem coupled with generalized upper bound (GUB) constraints. Beaujon et al. [19] also reported a MIP formulation designed to select projects for inclusion in a R&D portfolio, this model taking the form of a MKP with other generalized constraints. Capital budgeting appeared to be an on-going management challenge for not-for-profits hospitals and multihospital healthcare systems in the United States. In this context, a financially oriented capital budgeting framework is developed in [108] which uses a MKP formulation. The MKP has been also introduced to model problems including cutting stock [70], loading problems [20,165], investment policy for the tourism sector of a developing country [65], allocation of databases and processors in a distributed data processing [63], delivery of groceries in vehicles with multiple compartments [25] and approval voting [170]. More recently, the MKP has been used to model the daily management of a remote sensing satellite like *SPOT*, which consisted in deciding every day what photographs will be attempted the next day [179].

The MKP is also a subproblem of many general integer programs. By example, the solution of multicommodity network optimization problems with general step cost functions led to solve at each iteration a subproblem which can be converted into a MKP coupled with multiple choice constraints [62]. Moreover, the multiknapsack structure 'naturally' appeared in several interesting extensions of the MKP. In capital budgeting with multiple criteria and multiple decision makers, the use of both an analytical hierarchy process and an integer programming framework led to a two-phases method in which the second phase need to solve a MKP instance [112]. Cappanera and Trubian [24] introduced another interesting extension of the MKP, in which there are greater-than-equal inequalities, called *demand* constraints, besides the standard knapsack constraints, and the objective function coefficients are not constrained in sign. According to the fact that identifying feasible solutions is a hard task in this case, they proposed a two-stage tabu search based procedure which did not really take into account the underlying MKP structure.

In mathematical programming studies, the bidimensional case of the MKP appeared as a promising subproblem. This model was encountered with a polyhedral approach when $(1, k)$-configurations were used to enforce LP-bounds [3,145], and also in Lagrangean decomposition techniques, where each knapsack-like constraint can be associated with GUB constraint, or a cardinality constraint related to the variables fixed at 1 in any optimal solution, to strengthen dual upper bounds [59,133].

Finally, the renewed interest in the research community in computational integer programming has intensified, during the last decades, the use of MKP benchmarks. On the one hand the nonnegativeness and the density of the constraint matrix A, on the other hand the nonexistence of special constraints such as generalized upper bounds, special-ordered sets and plant-location constraints, distinguish this problem from the 0–1 linear programming problem. These hypotheses are fundamental because it has been shown that the existence of special constraints is essential to derive efficient methods for solving large-scale 0–1 linear problems (see for example [43]). Hence, suitable numerical experiments for testing any new approach to solve pure 0–1 linear programs cannot get away from it.

Previous surveys on the MKP can be found in Gavish and Pirkul [64], Fréville [52], Fréville and Plateau [59], Chu and Beasley [28] and Osorio et al. [144]. In this paper, we give a broader insight into the main results published in the literature with emphasis on recent publications. Section 2 presents exact methods and reduction techniques. We also discuss testing and implementation issues. Section 3 discusses theoretical results. In Section 4, we review heuristics for finding feasible solutions.

2. Exact methods

It is instructive to recall the history of the development of exact methods at the same time for the MKP and the single constraint version. The 0–1 knapsack problem has aroused great interest in the research community. The development of exact algorithms began several decades ago. The early effective procedures emerged in the 1980s [11,47,131]. They preceded a variety of solution methods including dynamic programming and its variants, a branch-and-bound network approach as well as special enumeration technique and reduction schemes. In their famous book [132], Martello and Toth reported experiments with algorithm MT2 solving uncorrelated and weakly correlated instances with up to more than 100,000 variables. Several recent and significant advances tackle the case of strongly correlated instances which remained very difficult to solve. These new approaches involve the hybridization of dynamic programming and branch-and-bound [153], the use of expanding core subproblem [149] and upper bounds obtained by adding valid inequalities on the cardinality of an optimal solution constraint [133], the combination of a new dynamic programming recursion and an additional cardinality constraint [134,150] and can effectively solve large problems of these type as well as other hard classes. Besides, several effective special-purpose codes are available to practitioners and researchers of the operations research community (by example among others FPK79 [47], MT2 and MTR [132], COMBO [134]).

2.1. Review

Though the MKP is a straightforward generalization of the single case, the situation is quite different when several constraints are taken into account. The development of exact algorithms for the MKP began during the sixties. Gilmore and Gomory [70] gave one of the early references which outlines a dynamic programming algorithm. Always within the dynamic programming stream, Green [80] proposed extensions to the method of Gilmore and Gomory, and Weingartner and Ness [184] developed another approach embedding heuristics. Marsten and Morin [128–130] combined dynamic programming and branch-and-bound approaches for solving the MKP, including among others the introduction of low time consuming heuristics and LP bounds. The elimination of irrelevant states during the dynamic programming computation has been also investigated by Ibaraki [94] and Isaka [98]. These hybrid methods enhance the basic dynamic programming approach and bring out interesting new ideas. However, due to the excessive space requirements as the single constraint case, instances can be solved only for small values of n and the capacities b_i. Consequently, neither of them are an effective self-containing solution for the MKP.

Other special approaches have tried to take advantage of the special structure of the MKP. Cabot [23] suggested an enumeration technique based on the Fourier–Motzkin elimination method and Thesen [174] presented a recursive branch and bound algorithm. Unfortunately, both methods have produced only scanty results. Soyster et al. [167] obtained a little bit more significant results with an iterative scheme. Starting with the solution of the LP relaxation, the basic variables are fixed to their values 0 or 1 and the remaining subproblem corresponding to the fractional variables is solved through implicit enumeration. If the feasible solution is not proved to be optimal, a disjunctive cut is added, which cuts off the current feasible solution and allows to strengthen the next LP-relaxation. However, convergence occurs only when the duality gap is small, because of numerical instability.

The seminal papers of Balas [6] and Glover [71] started the development of implicit enumeration techniques to solve 0–1 linear programs (see also [7,68]). Lemke and Spielberg [116], Trauth and Woolsey [178] and Breu and Burdet [22] investigated among others the computational effectiveness of several optimizing 0–1 codes based on these techniques. The ability of these methods for solving MKP instances exactly remained rather limited. By example, Wyman [185] formulated decision rules for choosing between optimal and heuristic algorithms by comparing a specific-purpose greedy heuristic with the code RIP30C [69,163]. This optimizing 0–1 code adds the features of surrogate constraints and an embedded linear program to accelerate the implicit enumeration method of Balas [6]. The reported results concerned randomly generated and uncorrelated MKP instances with a size which does not exceed 45 variables and 30 constraints. So far, implicit enumeration based branch-and-bound methods have not been competitive with other more significant approaches discussed further on.

Shih [165] designed the first linear programming-based branch and bound method which takes advantage of the special structure of the MKP. The estimation of an upper bound and the branching rule at any node are based on the information provided by the solutions of the LP relaxations associated to each of the m single-constraint knapsack problems. Shih reported computational experiments with a group of thirty randomly generated and uncorrelated problems with 5 constraints and 30–90 variables, and showed that the solution time of the improved Balas algorithm can be reduced by a factor ten in this way. The main drawbacks of this approach are first its excessive space requirements, second its inability to solve problems with tight resource constraints [64].

Lagrangean and surrogate relaxation based branch-and-bound methods have also been developed, but with varying results. Crama and Mazzola [30] gave an important result on the improvement in the bound that can be realized with these relaxations. They showed that for any $\lambda,\ \mu \in \Re^m_+$, the composite relaxation introduced by Greenberg and Pierskalla [71]:

$$z_{\mathrm{CR}}(\lambda,\mu) = \sum_{i\in M}\lambda_i b_i + \max\left\{ \sum_{j\in N}\left(c_j - \sum_{i\in M}\lambda_i a_{ij}\right)x_j \,\middle|\, \sum_{j\in N}\left(\sum_{i\in M}\mu_i a_{ij}\right)x_j \leqslant \sum_{i\in M}\mu_i b_i,\ x_j\in\{0,1\}, j\in N \right\}$$

is such that $z_{\mathrm{LP}} - \max_{j\in N}\{c_j\} \leqslant z_{\mathrm{CR}}(\lambda,\mu)$ and $\frac{1}{2}z_{\mathrm{LP}} \leqslant z_{\mathrm{CR}}(\lambda,\mu)$. This result states that the composite dual, and consequently the Lagrangean dual (with more than $m-1$ relaxed constraints) and the surrogate dual, cannot improve upon the bound z_{LP} obtained from the LP relaxation by more than $c_{\max}$, the largest objective-function coefficient.

Early investigations with Lagrangean multipliers in solving the MKP began with the paper of Lorie and Savage [120]. They proposed a kind of Lagrangean heuristic for 0–1 integer programming, in which all of the constraints were relaxed in the objective function. Their approach was later formalized by Everett [44] and extended by Kaplan [101]. Nemhauser and Ullman [140] showed that with respect to this relaxation, the problem of finding an optimal set of multipliers is equivalent to solving the dual of the LP relaxation. Stimulated by the successful application of lagrangean duality in solving structured combinatorial optimization problem [90,91], Etcheberry et al. [42] reported numerical experiments with two Petersen test problems [146] of small size by using an implicit enumeration approach where Lagrangean relaxation and subgradient optimization replaced linear programming. Barcia [12] and Barcia and Holm [13] presented an iterative algorithm, called ‘Bound Improving Sequence Algorithm’, based on Lagrangean relaxation. A decreasing sequence of upper bounds on the optimal value z^* is generated by adding cuts $cx \leqslant \mathrm{UB}$, where UB is an upper bound of z^*. The main drawback of this approach is that several very hard subset-sum problems need to be solved exactly to ensure finite convergence. In conclusion, even if elegant results and effective algorithms have emerged with Lagrangean duality for solving both 0–1 linear programs, particularly for problems for which the duality gap is very small or null, and 0–1 knapsack problems, by adding valid inequalities on the cardinality of an optimal solution and relaxing it in a Lagrangean fashion, it seems that a Lagrangean relaxation framework is not appropriate to tackle the simple and homogeneous structure of the MKP.

The surrogate strategy introduced by Glover [71] replaces the original constraints by a single new one, called a surrogate constraint. Greenberg and Pierskalla [81] provided the first major treatment of surrogate constraints in the context of general mathematical programming. The studies by Glover [72,73], Karwan and Rardin [102] and Dyer [38] complete the bulk of the research done in this area. Search procedures to find surrogate and composite multipliers were proposed by Karwan and Rardin [102–104], Karwan et al. [105], Dyer [38] and Glover [76] for general integer programs. Specific procedures were designed for the bidimensional 0–1 knapsack problem [53,58,85], able to find the optimal dual solution within a finite number of iterations, practically independent of the number of variables.

Although more effort is typically required to calculate the bounds, surrogate relaxations appear more useful for solving the MKP than Lagrangean relaxation. Gavish and Pirkul [64] developed a branch-and-bound procedure embbeding new approximate algorithms for obtaining surrogate bounds and rules for reducing problem size. As a matter of fact, they used a LP relaxation of the surrogate dual to avoid the solving of 0–1 knapsack problems and to lead to low solution times. In this way the LP bound was not improved, but attractive results were obtained. More precisely, they showed that their method was significantly faster than Shih's method and a general integer programming package called Sciconic/VM [17] by testing a group of problems with size up to 80 variables and 7 constraints. Sikorski [166] reported also numerical experiments with a branch-and-bound method based on surrogate duality. Fréville and Plateau [60] investigated the use of integer surrogate relaxations for solving the more simple case $m = 2$. Particularly, a specific and efficient preprocessing phase is designed, completed with an enumerative phase if needed. Computational experiments with several sets of randomly generated and correlated instances up to $n = 750$ showed that the procedure compares favorably with Gavish and Pirkul procedure and can provide for the bidimensional case a competitive alternative to LP-based strategies.

Constraint programming techniques integrated into integer programming is in an on-going research phase for solving mixed-integer programming problems. Preliminary work by Oliva et al. [143] in the context of MKP deals with the use of reduced costs to identify a set of constraints among which at least one must be satisfied to find a better solution than the current one. A constraint programming solver uses these new constraints to enhance the exploration of each node of the branch-and-bound search tree developed by CPLEX by reducing the feasible domain. Numerical results, reported in the bi-dimensional case on medium size (n up to 100), show both a decreasing number of explored nodes and an increasing CPU time. It is too early to make comments about the potential of such an approach. However, these results remain moderate and much more work must be done to assert the efficiency of constraint programming.

2.2. Commercial softwares and data sets

The best success for solving the MKP has been obtained recently with branch-and-cut algorithms embedding effective preprocessing. Moreover, widely used solvers including CPLEX [29], OSL [97] and XPRESS-MP [33] are available as well as systems such as MINTO [139] which have been built to provide an easier-to-use interface for the development of a special-purpose optimizer (see [100] for a comprehensive exposition). Although many impressive results have been obtained in the last decade for solving problems with thousands of integer variables or even more, it seems that the MKP remains rather difficult to handle when an optimal solution is wanted.

A common approach to study the efficiency of an approximate or exact method is to compare its performances with other optimizers on standard sets of test problems. The OR-Library [18], created by Beasley in 1990, [1] is a collection of integer programs that contains, among others, MKPs. The first set is compose of 56 small and uncorrelated instances, collected from earlier papers by Fréville and Plateau [56] and consisting of $m = 2$–30 and $n = 6$–105. This set is easy to solve. A great part of the optimal solution were contained in the early works of Shih [165] and Gavish and Pirkul [64], and more recently, all the problems were solved to optimality in less than 3 seconds of CPU time by CPLEX v6.5.2. [144].

The second set is a collection of 270 more difficult instances [28], generated using the procedure suggested by Fréville and Plateau [59]. Several experimental studies conducted with the single constraint case [132,157,187] and the multidimensional case [59,60] showed that correlation between the objective function and constraint coefficients contributes to the problems hardness. The constraint coefficients a_{ij} are drawn from the uniform distribution $U(0, 1000)$. The profit coefficients c_j are correlated to a_{ij} and generated as follows: $c_j = \sum_{i \in M} a_{ij}/m + Ku_j$ where u_j is drawn from the continuous uniform distribution $U(0, 1)$ and K is the correlation value. The parameter K is set to 500. Decreasing values of K increase the correlation and consequently, degrade solution procedure performance. The number of constraints was set to 5, 10 and 30, and the number of variables was set to 100, 250 and 500. For each m–n combination, thirty problems are generated and the constraint capacities b_i are set using the relation $b_i = \alpha \sum_{j \in N} a_{ij}$ where α is a tightness ratio: $\alpha = 0.25$ for the first ten problems, $\alpha = 0.50$ for the next ten problems and $\alpha = 0.75$ for the remaining test problems. Osorio et al. [144] observed that CPLEX v6.5.2., without modifying its default parameters, could solve only 95 instances to optimality in less than 3 hours and a memory size of 250 Mb. As a matter of fact, it is important to notice that in the rest of the problems, CPLEX usually terminated because the memory size was exceeded by the tree expansion. However, even if early termination occurs, CPLEX obtained lower bounds of good quality. For example, the average value is greater than the one obtained with the genetic algorithm developed by Chu and Beasley [28] within comparable CPU time requirements.

Following the empirical study in [92,113] of the effects of constraints coefficient generation on the performance of solution procedures, Osorio et al. [144] proposed a somewhat different problem generator. The constraint coefficients a_{ij} are drawn from the exponential distribution: $a_{ij} = 1.0 - 1000 \log(U(0, 1))$. It has been also observed [92,148] that the constraint slackness setting is a very significant factor influencing the size of the $z_{\mathrm{LP}} - z^*$ gap. Problems with tighter constraints tend to have larger $z_{\mathrm{LP}} - z^*$ gap values and then are more difficult to solve. A first set of 30 'difficult' problems consisting of 100 variables and 5 constraints was generated with a tightness ratio equal to 0.25. They observed that CPLEX v6.5.1 could solve only 19 instances to optimality in less than 3 hours with 128 Mb in RAM. A second set consisting of instances up to $n = 500$, 5 constraints and various tightness ratios leads to the same conclusion, i.e. these problems cannot be solved in a reasonable amount of CPU time and memory with CPLEX v6.5.2.

Finally, two other data sets have been proposed by Glover and Kochenberger [77]. [2] They are correlated and uniform randomly generated. The first set contains 24 fairly difficult problems in size up to 500 variables and 25 constraints. Optimality has been proved for the first seventeen instances. The best known feasible solutions for the last seven problems are given in Vasquez and Hao [180]. The second set includes still more difficult instances with n up to 2500. As far as we know, CPLEX cannot tackle these instances, even with its last enhanced versions. Therefore, it appears that the MKP continues to be a challenging problem for commercial ILP solvers.

[1] OR-Library is available at http://mscmga.ms.ic.ac.uk/info.html.

[2] Available at http://hces.bus.olemiss.edu/tools.html/.

2.3. Other implementation issues

As CPLEX is a general-purpose mixed integer programming solver, enhanced versions of CPLEX, or of any commercial solver, could be created by embedding preprocessing tools which held advantage of the special structure of the MKP.

Preprocessing techniques play a fundamental role in the development of efficient integer programming methods. Basic techniques try, among other things, to fix variables, to identify infeasibility and constraint redundancy, to tighten the LP-relaxation by modifying coefficients and by generating strong valid inequalities. Savelsbergh [160] gave a comprehensive survey of reduction techniques for 0–1 and mixed integer programming problems (see also [84,116]).

The seminal papers of Plateau [152] and Fayard and Plateau [46] introduced the main ideas to reduce the size of MKP instances by fixing variables and by eliminating redundant constraints. By example, the fixation of variables for any instance P needs the knowledge of a good lower bound $z_H = c\underline{x}$ associated with a feasible solution $\underline{x}$, and lies on the following basic property: for any $j \in \{1, \dots, n\}$ and for any $\varepsilon \in \{0, 1\}$, if $z(P|x_j = \varepsilon) \leqslant z_H$ then either $x_j = 1 - \varepsilon$ in any optimal solution of P, or $(P|x_j = 1 - \varepsilon, cx > z_H)$ has no feasible solution and $\underline{x}$ is optimal. Then, by considering several Lagrangean and surrogate relaxations of the MKP, more and more tigh upper bounds of $(P|x_j = \varepsilon)$ are computed with increasing complexity to achieve the inequality $z(P|x_j = \varepsilon) \leqslant z_H$. These rules extended the well-known fixing property based on the reduced costs associated to the LP-relaxation. Particularly, one of these rules brought into play the additivity of the reduced costs coupled with the set of *preferred* Lagrangean relaxations. These reduction rules have been tested intensively in order to identify their capabilities, both in the general case [59] and in the bi-dimensional case [60]. The numerical results showed that the effectiveness of this kind of size reduction fluctuated with the structure of the instances. A small number of constraints furthers the reduction effect, as well as uncorrelated and randomly generated entries generated with the uniform distribution.

A recent study developed a dynamic programming based reduction framework in front of a CPLEX [4]. Two sequences of upper and lower bounds are generated by solving LP-relaxations and by using dynamic programming respectively. Their comparison allows either to prove that the best feasible solution obtained is optimal, or to fix a subset of variables to their optimal values. Computational experiments with a large set of large-scale instances showed that the reduction framework is able to reduce the CPU time of CPLEX v7.0. However, the procedure failed to reduce the size in some cases, by example with the difficult instances of Glover and Kochenberger [77], but at least, good feasible solutions are obtained very quickly.

The recent study by Osorio et al. [144] focused on the generation of logic cuts, and also allowed to fix variables to zero by using the above property based on the reduced costs associated to the LP-relaxation. Two types of logic cuts are generated by hybriding surrogate analysis [78] and constraint pairing [21]. Many numerical experiments are conducted with the previous data sets and showed that CPLEX v6.5 performed much better on average when augmented with this procedure, the main progress being obtained thanks to the logic cuts.

Parallelization of tree search algorithms has received wide attention in the computer science community, and has focused on scalable parallel task scheduling and distribution strategies. Besides, one can remark that specific developments have been carried out for the MKP. Plateau and Roucairol [154] and Mans [127] presented a parallel framework, including the search of an initial feasible solution, size reduction based on the additivity of the reduced costs, and a terminal branch-and-bound phase. A linear speed up has been obtained with randomly generated problems and runs performed on a CRAY-2. However, no impressive results have been reported in the literature as far as we know.

On the other hand, Johnson et al. [100] mentioned that parallel branch-and-cut and parallel branch-and-price seem to have great potential in mixed integer programming. This new research area is certainly a promising way to treat MKP instances with a large number of variables and a reasonable number of constraints.

3. Theoretical analysis

Besides numerical testing, theoretical analysis, first helps us to understand the huge increase in difficulty when moving from a single constraint to more than two, and second, that uncorrelated instances randomly generated with the uniform distribution are rather easy to solve.

3.1. Approximation algorithms

As the single case, the MKP is *NP-hard* but not *strongly NP-hard*; indeed, no problem that is *strongly NP-complete* can have a *pseudo polynomial algorithm* unless $P = NP$, and it is well-known that dynamic programming provides for both problems such schemes requiring at most $O(n\prod_{i=1}^{m}(b_i + 1))$ computation.

It is also well-known that *polynomial approximation schemes* exist for both cases $m = 1$ and $m > 1$. The first polynomial approximation scheme is due to Shani [164] for the 0–1 knapsack problem. The case $m > 1$, with m considered to be a *fixed constant*, was analysed by Chandra et al. [26] with constraint (2) replaced by x_j nonnegative integer. Magazine and Oguz [123] extended their method to the MKP. Frieze and Clarke [61] gave an alternative approach, which is a straightforward extension of the idea of Shani. Moreover, polynomial approximation schemes link with worst-case analysis, which provides a guarantee on the maximum amount that a heuristic algorithm will deviate from optimality for any instance. The *worst-case performance ratio* is the largest r $(0 \leqslant r \leqslant 1)$ for which $z_H(I)/z^*(I) \geqslant r$ holds for any instance $I \in$ MKP, where $z_H(I)$ denotes the value provided by the heuristic algorithm H applied to the instance I. This ratio is usually not predictive of average performance.

Polynomial approximation scheme allows to exhibit for the MKP polynomial time heuristic algorithm for any r arbitrarily close to 1 [61,123]. As the single case, these methods involve partial enumeration of subsets of items but greedy packing of items is replaced by LP-rounding. By example, for any $\varepsilon > 0$, the heuristic $E(k)$ in [60] is parametrized by an integer $k = \min\{n, \lceil\frac{m(1-\varepsilon)}{\varepsilon}\rceil\}$ and has a worst-case performance ratio of $r = 1 - \varepsilon$. For $S \subseteq N = \{1, 2, \ldots, n\}$, let denote $T(S) = \{t \in N \setminus S | c_t > \min\{c_j | j \in S\}\}$ and $z_{\lfloor LP \rfloor}(S)$ the value of a LP-rounding solution of the MKP of capacity $b - \sum_{j\in S} A^j$ and restricted to the items $j \in N \setminus S \cup T(S)$. The kth level partial enumeration algorithm obtains a solution by solving

$$z_{E(k)} = \max\left\{\sum_{j\in S} c_j + z_{\lfloor LP \rfloor}(S) | \sum_{j\in S} a_{ij} \leqslant b_i,\right.$$
$$\left. i = 1, \ldots, m, |S| \leqslant k\right\}$$

and the problem is solved by enumeration of all sets S of cardinality k or less. If L is a measure of the length of space needed to describe the problem, the time complexity in the worst case is $O(n^{k+5}L^2 \log(nL) \log\log(nL))$ if Khachian's algorithm is used to evaluate the value $z_{E(k)}$.

For the 0–1 knapsack problem, Ibarra and Kim [96] gave the first *fully polynomial approximation scheme* based on dynamic programming techniques, which runs in $O(n/\varepsilon^2 + n \log n)$ time, i.e. in polynomial time both in the number of variables n and in the inverse of the worst case performance ratio ε. Further improvements are found in [114,123]. The main difference between the cases $m = 1$ and $m > 1$ emerges at this point. Unless $P = NP$, Gens and Levner [66] and Korte and Schrader [110] have proven that no fully polynomial approximation scheme exists as soon as the number of constraints is greater than or equal to 2 (see also [122]). Hence this fundamental result gives an insight into the increase in difficulties due to the change of a single constraint by multiple constraints.

3.2. Probabilistic analysis

A lot of papers have tackled the probablistic analysis of the MKP during the 1990s. Most of the results investigated the dependence of the solution value z^* on the constraint capacities b_i and on the number of variables n. Asymptotic but also analytical results are given under various assumptions on how the coefficients are generated.

An optimal basic solution $\bar{x}$ to the LP relaxation (constraint (2) replaced by $0 \leqslant x_j \leqslant 1,\ j \in N$) has at most m basic variables that are fractional. Rounding down these values yields a feasible solution $\lfloor \bar{x} \rfloor$ such that $z_{\mathrm{LP}} - mc_{\max} \leqslant z_{\lfloor \mathrm{LP} \rfloor} = c\lfloor \bar{x} \rfloor \leqslant z^* \leqslant z_{\mathrm{LP}}$. This *LP-rounding heuristic* performs usually poorly. However, if it is assumed that the coefficients c_j remain bounded as n is increasing, the asymptotic equivalences $z^* - z_{\lfloor \mathrm{LP} \rfloor} = \mathrm{O}(n)$ [3] and $z_{\mathrm{LP}} - z^* = \mathrm{O}(n)$ follow easily. Frieze and Clarke [61] studied more precisely the asymptotic properties of the *LP-rounding heuristic*. When it is assumed that all coefficients c_j and a_{ij} are nonnegative real numbers drawn independently from the uniform [0,1] distribution and $b_i = 1$ for $i = 1, 2, \ldots, m$, they showed that $(((z^* - z_{\lfloor \mathrm{LP} \rfloor}))/z^*) \leqslant \varepsilon$ with probability tending to 1 as $n \to \infty$, provided ε is $\mathrm{O}(n^{-\alpha})$ where $\alpha < (1/m + 1)$.

With the same realizations of random variables c_j and a_{ij}, Dyer and Frieze [39] generalized two results stated at first for the one dimensional case. The first result gave an asymptotic characterization of the expectation of the gap $z_{\mathrm{LP}} - z^*$. More precisely, if the capacities b_i grow proportionally with the number n of items ($b_i = n\beta_i$ where $0 < \beta_i < 1/2$ are fixed for $i = 1, 2, \ldots m$; it can be remarked that the ith constraint would tend to redundant if $\beta_i > 1/2$), then $E(z_{\mathrm{LP}} - z^*) \leqslant \alpha(\log^2 n/n)$ with a constant $\alpha > 0$ depending only on m and $\beta = \min\{\beta_1, \ldots, \beta_m\}$. In addition, they proved that for any $\varepsilon > 0$, there exists an approximation algorithm based on the LP relaxation, which runs in $\mathrm{O}(n^{f(\varepsilon,m,b)})$ for some function $f(\varepsilon, m, b) > 0$ and with probability at least $1 - \varepsilon$ solves the MKP.

Several papers investigated the question, raised by Frieze and Clarke [61], of computing the asymptotic value of the random variable z^* for fixed m. Mamer and Schilling [125] and Schilling [161,162] gave a main insight according to the assumptions made in [61]; they proved that z^* grows like $(m + 1)(n/(m + 2)!)^{1/m+1}$ with a probability one as n goes to infinity. However, the case with all the capacities b_i equal to 1 is a rather artificial model, and Szkatula [171–173] proposed a generalization of the model which allowed the right-hand-sides b_i to be functions of n with quite different values. These results allowed to observe how the asymptotic behavior of z^* is influenced by the coefficients n, m, b_i, and more indirectly, by c_j, a_{ij}.

Meanti et al. [136] obtained an important result by using Lagrangean relaxation technique. Under a stochastic model a little more general than the one considered in [39], they proved that z^*/n converges with a probability one to $L(\lambda^*)$ as n goes to infinity and m remains fixed. $L(\lambda^*)$ is a function of the right hand sides b_i and is defined implicitly by minimization of $L(\lambda)$, a kind of Lagrangean relaxation expectation. λ^* is the unique minimizer of $L(\lambda)$ over a compact set of multipliers λ. However, explicit results were presented only for the case $m = 1$ and $m = 2$. Geer and Stougie [67] established a rate of convergence for the result of Meanti et al. using results from the theory of empirical processes. Specifically, they proved that $(n/\log \log n)^{1/2}|(z^*/n) - L(\lambda^*)| = \mathrm{O}(1)$ with probability equal to one. Following the Lagrangean relaxation approach of Meanti et al., Fontarani [50] obtained an analytic result with a different stochastic model where the a_{ij} are always independent random variables uniformly distributed over [0, 1], but $c_j = 1$ for all j and $b_i = \beta n$ for all i. For any finite number of constraints, the optimal value z^* is given explicitly as function of the knapsack capacities.

Averbakh [5] also studied the probabilistic properties of the Lagrangean dual of the MKP. Under the assumption that coefficients are independently distributed, Averbakh established several results of asymptotic type dealing with the probability of existence of *ε-optimal* ($\sum_{j=1}^{n} c_j x_j \geqslant (1 - \varepsilon)z^*$) and *δ-feasible* ($\sum_{j=1}^{n} a_{ij} x_j \leqslant (1 + \delta)b_i, i = 1, \ldots, m$) Lagrangean function generalized saddle points.

An other interesting approach was presented by Rinnooy Kan et al. [158] in terms of a *primal greedy algorithm* (see Section 4.1.1). Building on the results of Meanti et al., they showed under the same assumptions and with the optimal dual

[3] $\mathrm{O}(n)$ denotes as usual some function that tends to zero as $n \to \infty$.

multipliers λ^* as weights w, that the random variable denoting the solution value of the greedy algorithm is asymptotically optimal with probability one.

Finally, it can be quoted that, based on the "replica formulation" of disorder physics [51], the asymptotic behavior of the optimal value z^* has been also investigated in recent years [50,111]. Their main interest was to provide analytic characterization of z^* but under quite restrictive assumptions over the stochastic model.

4. Heuristics

Even if recent advances based on the methodology of branch-and-cut have made possible the solution of middle size MKP instances, heuristic methods remain a competitive alternative, particularly when the number of constraints is large.

As mentioned before, a lot of papers have addressed heuristic approaches to the MKP, among which one can find again all the heuristic approaches introduced in combinatorial optimization. It is then a difficult task to suggest a classification. Nevertheless, we aim to roughly propose a grouping of heuristics into greedy algorithms, mathematical programming approaches and metaheuristics, in addition the more isolated attempts including specific local search [147], multistart strategies [40,155,159] and branch-and-bound early termination [64]. Wolsey gave a major treatment of heuristic analysis based on partial enumeration with the MKP taken as example [186]. Particularly, he showed how certain heuristic methods can be integrated into enumeration schemes to produce branch-and-bound algorithms whose worst case behavior steadlily improves as the enumeration develops.

Among this huge number of papers, it is important to underline that some 'good ideas' have emerged during the last three decades. Greedy-like assignment, LP-based search, surrogate duality information, local search allowing infeasible and worsening solutions, embedding heuristic in a reduction framework, are the ingredients of the most efficient procedures.

4.1. Greedy algorithms

Greedy, alternatively called 'myopic' or 'constructive/destructive', algorithms are fast (polynomial time complexity), and generally simple to implement. After the initial work of Edmonds dealing with matroid intersection [41], one of the earliest effective uses of greedy algorithms may have been the enumerative algorithms for simple plant location in Spielberg [168,169] and for the 'weakly linked' problems in Guignard and Spielberg [83].

4.1.1. Review

The early approaches extended the successful use of the 'bang-for-buck' ratios employed for solving the single constraint knapsack problem, and which are simply defined as the ratios of the profits and the resource coefficients. In the multidimensional case, the items are selected according to ratios $c_j / \sum_{i=1}^{m} w_i a_{ij}$, where $w = (w_1, w_2, \ldots, w_m)$ are given nonnegative weights. The first proposal stated by Senju and Toyoda [163] was a *dual* heuristic starting with the all-ones solution and setting the variables to zero one at a time according to increasing ratios until feasibility requirements are satisfied. By contrast, a class of *primal* methods was developed which started from the origin and set variables to one according to decreasing ratios until no more variables can be added without violating the constraints [109,121, 177].

At this point, one can underline that the empirical study conducted by Hill and Reilly [92] on the influence of correlation measure and correlation structure in the bidimensional case, reports a rather surprising result. In spite of its rough design, the primal greedy algorithm by Toyoda [177] performs better than CPLEX (v2.1 utilized in a depth-first search, branch-and-bound mode) for instances with many solutions with near-optimal values.

Next, dual multipliers have been employed in the definition of the selection rules to design more competitive greedy algorithms. Magazine and Oguz [124] combined the Senju and Toyoda 's dual algorithm with a Lagrangean relaxation approach, which allows fixing variables to their values assigned in all optimal solutions. Their work has been extended by Volgenant and Zoon [181]. Following the seminal work of Glover [74], Fréville and Plateau [54,55] gave three solution methods using surrogate constraints, and other concepts such as accelerated fixing (more than one variable fixed at a time), noising strategy and strongly determined variables. Pirkul [148] developed a more straightforward generic approach embedding a descent procedure to determine the surrogate constraints. With instances up to $n = 200$ and $m = 20$, Pirkul proved that this greedy procedure was generally faster than one of the major LP-based heuristics, the so-called Pivot and Complement heuristic described below, and generated solutions were similar in terms of solution quality. Finally, Hanafi et al. [88] proposed a two-stage multistart algorithm which incorporates different heuristic principles in a flexible fashion. Starting from a set of random feasible solutions, the first stage performed different local searches such that the threshold accepting [36,37] and the noising method [27]. An additional stage based on repeated greedy steps tries to improve the current feasible solution.

An advantage of these methods is that they enabled the determination of an upper bound to z^* to control the quality of the lower bound. Moreover, the lower and upper bounds have been combined with reduction techniques to design efficient preprocessing procedures [59]. Within this context, Lee and Guignard [115] considered a multistage procedure tuned with a few parameters which control the tradeoff between solution quality and computation times, and whose values are set by the users. Numerical results are reported with 48 test problems with 5–20 constraints and 6–500 variables, randomly generated using the same generation methods as in [9,99]. The solution found was on the average within 0.34% of the optimum and the computation time was the shortest compared with greedy algorithms [124,177] and the Pivot and Complement heuristic [9].

Beyond its respectable merit in terms of the quality of solutions obtained and the amount of computational time consumed, another major contribution of the greedy concept concerned the metaheuristics. Indeed, DROP-ADD moves (Fig. 1) and repair operators (Fig. 2) involving greedy-like assignments as shown in the following examples, have been used in the more efficient methods developed in the areas of tabu search [31,32,86] and genetic algorithms [28,175].

4.1.2. Worst-case analysis

The first results extend the ones obtained in the single case and deal with the well-known *primal greedy algorithm* described above. Following an example given in [48] for the single case, the series of instances $\{n = 2, m = 2, c_1 = 1, c_2 = k, a_{11} = 1, a_{21} = k, a_{12} = k, a_{22} = k + 1\}_{k=1,2,\dots,}$ show that the worst-case performance of this heuristic for any weights $w = (w_1, w_2) > 0$ is as bad as 0. Indeed, one has $z_G(w) = 1$ and $z^* = k$, so $z_G(w)/z^*$ can be arbitrarily close to 0.

<u>*DROP-ADD move*</u> (from a feasible solution x^{now} to a feasible solution x^{next})

> $x \leftarrow x^{now}$;
>
> choose $j^* = \arg\min \{ \frac{c_j}{A_{i^*j}} \mid x_j = 1, j = 1,\dots,n\}$ with i^* being a scarce resource suth that
>
> $i^* = \arg\min \{[b_i - \sum_{j=1}^{n} A_{ij} x_i]/b_i \mid i \in M\}$; $x_{j^*} = 0$; feasible = true ;
>
> <u>while</u> feasible <u>do</u>
>
>   choose $k^* = \arg\max \{ \frac{c_j}{uA^j} \mid x_j = 0, j = 1,\dots,n, j \neq j^*\}$ with $A^{k^*} + Ax \le b$;
>
>   <u>if</u> k^* exists <u>then</u> $x_{k^*} = 1$ <u>else</u> feasible = false <u>endif</u>
>
> <u>endwhile</u>
>
> $x^{next} = x$;

Fig. 1. DROP-ADD move.

Repair operator (from an infeasible solution x^{now} to a feasible solution x^{next})

$x \leftarrow x^{now}$; feasible = false ;
Let $I(x) = \{ i \in M \mid \Delta_i = \sum_{j=1}^{n} A_{ij} x_j - b_i > 0 \}$ be the subset of constraints violated by x ;
while not feasible do
    choose $j^* = \arg\min \{ \frac{c_j}{uA^j} \mid x_j = 1, j = 1,\ldots,n \}$; $x_{j^*} = 0$;
    update Δ ($\Delta = \Delta + A^{j^*}$) and I(x) ; if $I(x) = \emptyset$ then feasible = true endif
endwhile
$x^{next} = x$;
(by example, in Senju and Toyada [163], the multiplier u changes at each iteration by using $u_i = \Delta_i$ if $\Delta_i > 0$, $u_i = 0$ otherwise).

Fig. 2. Repair operator.

An interesting analysis of the class of *primal greedy algorithms* was conducted by Rinnooy Kan et al. [158]. They showed that the best greedy solution is not necessarily optimal, but that with weights chosen as the LP-dual multipliers λ^*, it performs always better than the LP-rounding heuristic, that is $z_G(\lambda^*) \geqslant z_{\lfloor LP \rfloor}$ (the two solutions are of equal quality in the worst case). They also pointed out that λ^* is not the optimal weight and studied the computational complexity of determining the optimal weight.

Dobson [34] established tight bounds on the worst-case behavior of dual greedy algorithms for the minimization case. Starting from the all-ones solution and using the weights $w = (1, 1, \ldots, 1)$, the items are deleted according to increasing ratio $c_j / \sum_{i=1}^{m} a_{ij}$ and coefficient reduction applies in the remaining columns A^j as soon as the coefficient a_{ij} becomes greater than the update constraint level b_i for any i. As the MKP can be easily transformed into a set covering problem $\min\{cx | Ax \geqslant b, x \in \{0,1\}^n\}$ by complementing the variable $x_j \leftarrow 1 - x_j$, the greedy algorithm of Dobson (GD) is nothing else than a dual greedy algorithm applied to the maximization version. Fischer [48] coined the problem that paradoxical dichotomies between minimization and maximization depend strongly on the use of the ratio z_H/z^* to measure performance. Thus, the worst-case analysis can be adapted to the maximization case by using another ratio to measure performance. Then, using the following performance measure $p_R(H) = z_R - z_H / z_R - z^*$ where $z_R = \sum_{j=1}^{n} c_j$ is a suitably reference value, one obtains that $p_R(\mathrm{GD}) \leqslant H(d)$, where $d = \max_{j=1,\ldots,n} \sum_{i=1}^{m} a_{ij}$ and $H(d)$ denotes the first d terms of the harmonic series $H(d) = 1 + \frac{1}{2} + \cdots + \frac{1}{d}$.

4.2. Mathematical programming

Most of the papers deal with integer/0–1 linear programs. As the LP-rounding heuristic, they often exploited the rudimentary idea that optimal or near-optimal integer solutions are close to the solution of the LP relaxation. But the use of a mathematical programming framework gave rise to many attractive developments.

Hillier [93] introduced several seminal ideas in his seminal paper. Hillier's algorithm is to our knowledge the first example of a *multistage* algorithm. The first phase identifies a path leading from the optimal LP solution to another nearby solution belonging to the integer feasible region. During the second phase, the algorithm moves along this path to identify a better feasible integer solution. This step is the first example of search strategy, called *path relinking*, a concept which was coined by Glover later [75]. Finally, the third phase realizes a *local search* which attempts to improve the current feasible solution by changing one or two variables at a time. Zanakis [188] showed with MKP instances of moderate size that Hillier's algorithm was more accurate than basic *primal/dual* greedy algorithms. In comparison with the complexity of the greedy methods, the embedded simplex LP accounting for almost 3/4 of the computing time appeared at the time as too much time consuming for solving large instances. Experimental results and others advances focused on how to construct efficient 'interior paths' are found in [45,95,99].

Pivot and Complement, developed by Balas and Martin [9], is probably the most well-known LP-based procedure for finding approximate solutions to general linear 0–1 programs (see [10] for the mixed integer programming case). The procedure starts off by solving the LP relaxation with a standard bounded variable simplex method and continues by performing a sequence of pivots aimed at putting bounded variables into the basis at a minimal cost. When this has been achieved a complementing phase tries to improve the 0–1 solution obtained in the pivoting. This procedure has performed well in practice on a variety of types and sizes of binary integer programs and has been embedded in several commercial codes. Thus it can be considered as a major reference. A further improvement was achieved by introducing an additional constraint [1]. Moreover, promising results have been also obtained for pure 0–1 linear programs by hybrids of tabu search with the Pivot and Complement heuristic. In Aboudi and Jornsten [2], the tabu search framework is superimposed on the Pivot and Complement heuristic used as a black box. On the other hand, tabu search is embedded within the Pivot and Complement heuristic [119]. Finally, Nediak and Eckstein [138] presented a pivot-based method related to the Pivot and Complement and Glover–Lokketangen tabu search methods. Their method attempts to round a fractional solution of the LP relaxation by a local minimization procedure for a concave merit function taking the value 0 at all integer-feasible points. This minimization is accomplished by simplex-like pivots starting, relying only on local gradient information until they reach a local optimum. Numerical results are reported only with 49 MIP test problems selected from the MIPLIB 3.0 collection, and no experiment is to date available with MKP instances.

Balas et al. [8] proposed a sophisticated local search, called OCTANE, in the integer neighborhood of the fractional LP-solution for solving pure 0–1 programs. A set of feasible solutions is generated by computing the intersection points of some facets of an octahedron, containing the solution x to the LP-relaxation, with the half lines originating at x and having selected directions. They showed that their rather complex method is also a competitive alternative to Pivot and Complement.

More recently, Plateau et al. [151] investigated a multistage method bringing into play metaheuristics and *interior point* methods. The first phase consists of a hybrid search combining an interior point method to generate fractional germ points, a local search to restore feasibility, and a cut generator to diversify the population of initial feasible solutions. The second phase performs a fixed number of *path relinking* runs between a set of pairs of solutions selected in the initial population. Preliminary comparisons with the Chu and Beasley algorithm indicate promising prospects for using interior point methods as a guide to enhanced local search, path relinking or scatter search [75].

A more user-friendly heuristic and easy to implement is proposed in Andonov et al. [4]. The main idea is the use of dynamic programming in a suitable way to get a feasible solution by successive improvements of the LP-rounding solution. Tested on all standard sets of the literature, this heuristic, embedded into a preprocessing framework with possible reduction of the problem size, is shown to be robust and very fast compared with the best tabu search approaches. Particularly, for the last seven instances of Glover and Kochenberger, with the optimal solutions unknown, the relative deviation to the best known feasible solution is less than 0.24% and the CPU time less than 0.1 seconds on a DEC Alpha workstation.

Oddly, as compared with the success of *Lagrangean heuristics* for solving many 0–1 programs of various characteristics [49], very few papers reported, to our knowledge, relevant experiences with perturbing optimal solutions to Lagrangean or surrogate relaxations, so as to obtain good feasible solutions. The first idea was developed in Fréville and Plateau [54,55]. The variables fixed at 1 in the solution of a perturbed continuous surrogate relaxation are fixed temporarily at 1, the others are fixed at 0. If the solution generated in this way is infeasible, a dual greedy algorithm sets free some of the variables fixed at 1 to move back into the feasible region. The variables remaining at 1 are fixed definitively, and the procedure starts again with the subproblem defined by the free variables. However, the quadratic time complexity limits its utilization to medium size instances. The work of Guignard et al. [82] is another attempt. Lagrangean decomposition was investigated in the bidimensional case with a limited success unfortunately. Indeed, the size of the lagrangean dual is equal to the number of variables and leads to overly expensive time requirements.

4.3. Metaheuristics

As for many others combinatorial optimization problems, the MKP has been intensively investigated within metaheuristics during the last decade. Most of the research has considered the MKP an interesting benchmark but has not exploited its specific properties sufficiently to state effective methods. However, several recent papers bringing into play relevant mechanisms have obtained the best results reported in the literature.

The first attempt was realized by Drexl [35] with *simulated annealing*. He introduced a special 2-exchange random move which maintains the feasibility of all solutions generated during the process. Battiti and Tecchiolli [14] introduced a penalty function which transforms the MKP into an unconstrained problem. They reported that SA performance varies greatly with the MKP instance and was worse than suitable *tabu search* implementations, which used different search strategies to overcome local optimality. Dueck and Scheuer [36] and Dueck and Wirsching [37] presented a close and deterministic version of SA, called *threshold accepting*, with rather better results than the SA version of Drexl.

Pioneer research in the field of *tabu search* for solving the MKP started with Dammeyer and Voss [31] who compared static and dynamic strategies for managing the tabu list. In particular, they showed that SA may be outperformed by a dynamic version of TS, called *Reverse Elimination Method*, in which feasibility is maintained along the process by using a multivariate DROP/ADD move [32,182]. In this context, recent extensions concerning the link between new dynamic rules and diversification and intensification strategies are found in Hanafi and Fréville [87]. Another tabu list dynamic management, called *Reactive Tabu Search*, has been tested with satisfactory performances by Battiti and Tecchiolli [15,16].

Other TS approaches have been designed for solving mixed 0–1 integer programs, in which infeasibility dealing with the integrality requirements is allowed during the process. In addition to the hybrids of TS coupled with pivot and complement mentioned above, Lokketangen and Glover developed a direct approach by making TS rely on a standard bounded variable simplex method as a subroutine [117]. Others advances for designing efficient TS mechanisms within this framework are found in [118].

However, it is undeniable that the best results have been obtained with methods which exploited in-depth the MKP's property that all the near optimal solutions lie in the boundary of the feasible domain. The first main idea was the use of *tunneling effect*. Glover and Kochenberger [77] obtained computational results of high quality over several test beds of large size up to 500 variables and 25 constraints. They presented a *strategic oscillation* scheme which alternates between constructive and destructive phases and drives the search to variable depths on each side of the feasibility boundary. Hanafi and Fréville [86] obtained competitive results with those of Glover and Kochenberger, by developing a TS approach which combines strategic oscillation with generalized greedy algorithms guided by surrogate constraints information and the state of the search. Vasquez and Hao [180] presented another hybrid and effective strategy. They introduced an additional constraint $\sum_{j=1}^{n} x_j = k$ on the cardinality of an optimal solution which allows pruning a subset of the feasible domain. LP-rounding solutions are computed for each suitable value of parameter k, and after that, a local TS using a dynamic tabu list is performed around these starting points (other efficient strategies are found in Glover [71], Fréville and Plateau [57] to compute suitable values of k). Until now, they reported the best solutions in quality with the difficult test beds of Glover and Kochenberger and Chu and Beasley. But the amount of computational time is rather high and that remains a handicap for solving very large instances exceeding a thousand of variables and more than ten constraints.

Evolutionary algorithms is another important stream of metaheuristics. The early papers have not successfully proved that *genetic algorithms* were an effective heuristic tool for the MKP. Khuri et al. [107] extended previous work for the single constraint knapsack problem [106]. They developed a GA with standard operators and a fitness function which penalizes the unfeasible strings. A similar study is given in Battiti and Tecchioli [14], which provided also comparisons with others metaheuristics. Thiel and Voss [175] showed that a standard GA using a direct search in the complete search space is not able to obtain good solutions for the MKP, except for small problems. Moreover, they investigated the combination of GA with tabu search and obtained promising results when applied to test problems of moderate size. Chu and Beasley [28] gave the first successful implementation of GA's by restricting the genetic algorithms to search only the feasible search space. Extended numerical comparisons with CPLEX (version 4.0) and other heuristic methods showed the robust behavior of the method for obtaining high-quality solutions within a reasonable amount of computational time. These results have been improved recently by incorporating local optimization to focus the search process on the boundary of the feasible region [79,156]. Finally, Haul and Voβ enhanced the performance of GA's by using surrogate constraints [89].

The use of *neural networks* to solve the MKP was pioneered by Battiti and Tecchioli [14] and Ohlsson et al. [142]. Through the reported numerical experiments, NN appeared to be of limited practical interest. In particular, due to the strategic choice of a penalty function which transforms the MKP into an unconstrained problem, NN tends to produce final solutions that violate constraints.

Parallelization of metaheuristics is a field in growing expansion (see Toulouse et al. [176] for a comprehensive survey). As parallel branch-and-cut and parallel branch-and-price methods, this research area is certainly a promising way to tackle very large-scale instances. Nevertheless, only very few papers have addressed the MKP. Niar and Fréville [141] presented a parallel implementation of TS using a synchronous communication scheme. Parallelism focused on the exploration of the solution domain, by maintaining different independent/dependent search paths. Numerical results showed that parallel cooperative search allows both reducing CPU time and setting dynamically strategic TS parameters.

5. Conclusion

We have tried to convey the message that passing from one to more than two constraints generates a significant gap in difficulties. Regarding exact solutions, the MKP has been studied less as a generalization of the knapsack than as a special case of 0–1 linear program. So, very few specialized procedures are available. They meet limited success, except in the bidimensional case where surrogate relaxations offer promising insight. On the other hand, the computational improvements of the 1990s, that have integrated heuristics, preprocessing and probing techniques, branch-and-bound and strong valid inequalities, lead to significant improvements in the exact solving. However, even with the more effective solvers, the largest sizes of instances for which exact solutions are given, do not exceed a few hundred variables as soon as the number of constraints increases. In contrast, and always due to its attractive 'progressive complexity', both linked to the number of variables and constraints, many special-purpose heuristic algorithms have been developed to provide competitive alternatives to branch-and-bound or branch-and-cut algorithms with a time limit. Enhanced versions of tabu search provide, to date, the best near-optimal solutions with the more difficult test beds. At the same time, less expensive CPU time requirements and robust heuristics are available, most of them based on mathematical programming foundations as LP-relaxations and dynamic programming.

Acknowledgements

I am indebted to Kurt Spielberg who read earlier version of the paper, providing valuable suggestions and pointing out several errors.

References

[1] R. Aboudi, A. Hallefjord, R. Helming, K. Jornsten, A note on the pivot and complement heuristic for 0–1 programming problems, Operations Research Letters 8 (1989) 21–23.

[2] R. Aboudi, K. Jornsten, Tabu search for general zero–one integer programs using the Pivot and Complement heuristics, ORSA Journal of Computing 6 (1994) 82–93.

[3] R. Andonov, Finding facets of the knapsack polytope which cut off a given point, Comptes Rendus de l'Académie Bulgare des Sciences (1987) 41–44.

[4] R. Andonov, S. Balev, A. Fréville, N. Yanev, A dynamic programming based reduction procedure for the multidimensional 0–1 knapsack problem, Working paper, University of Valenciennes, presented at FRANCORO III, Québec, Canada, May 2001. Forthcoming in European Journal of Operational Research.

[5] I. Averbakh, Probabilistic properties of the dual structure of the multidimensional knapsack problem and fast statistically efficient algorithms, Mathematical Programming 65 (1994) 311–330.

[6] E. Balas, An additive algorithm for solving linear programs with zero–one variables, Operations Research 13 (1965) 517–546.

[7] E. Balas, Discrete programming by the filter method, Operations Research 19 (1967) 915–957.

[8] E. Balas, S. Ceria, M. Dawande, F. Margot, G. Pataki, OCTANE: A new heuristic for pure 0–1 programs, Operations Research 49 (2001) 207–225.

[9] E. Balas, C.H. Martin, Pivot and complement—A heuristic for 0–1 programming, Management Sciences 26 (1980) 86–96.

[10] E. Balas, C.H. Martin, Pivot and shift—a heuristic for mixed integer programming, Working paper, GSIA, Carnegie Mellon University, USA, 1986.

[11] E. Balas, E. Zemel, An algorithm for large zero–one knapsack problems, Operations Research 28 (1980) 1130–1145.

[12] P. Barcia, The bound improving sequence algorithm, Operations Research Letters 4 (1985) 27–30.

[13] H. Barcia, S. Holm, A revised bound improvement sequence algorithm, European Journal of Operational Research 36 (1988) 202–206.

[14] R. Battiti, G. Tecchiolli, Parallel biased search for combinatorial optimization: Genetic algorithms and tabu search, Microprocessors and Microsystems 16 (1992) 351–367.

[15] R. Battiti, G. Tecchiolli, The reactive tabu search, ORSA Journal on Computing 6 (1994) 126–140.

[16] R. Battiti, G. Tecchiolli, Local search with memory: Benchmarking RTS, OR-Spektrum 17 (1995) 67–86.

[17] E.M.L. Beale, Branch and bound methods for mathematical programming systems, Annals of Discrete Mathematics 5 (1979) 201–219.

[18] J.E. Beasley, OR-Library: Distributed test problems by electronic mail, Journal of the Operational Research Society 41 (1990) 1069-1072 (http://mscmga.ms.ic.ac.uk/info.html).

[19] G.J. Beaujon, S.P. Martin, C.C. McDonald, Balancing and optimizing a portfolio of R&D projects, Naval Research Logistics 48 (2001) 18–40.

[20] R. Bellman, Dynamic Programming, Princeton University Press, 1957.

[21] G.H. Bradley, P.L. Hammer, L. Wolsey, Coefficient reduction for inequalities in 0–1 variables, Mathematical Programming 7 (1974) 263–282.

[22] R. Breu, C.A. Burdet, Branch and Bound experiments in zero–one programming, Mathematical Programming Study 2 (1974) 1–50.

[23] A.V. Cabot, An enumeration algorithm for knapsack problems, Operations Research 18 (1970) 306–311.

[24] P. Cappanera, M. Trubian, A local search based heuristic for the demand constrained multidimensional knapsack problem, Working paper presented at Congress Adaptive Memory and Evolution: Tabu Search and Scatter Search, Oxford, USA, March 2001.

[25] E.D. Chajakis, M. Guignard, A model for delivery of groceries in vehicles with multiple compartments and Lagrangean approximation schemes, VI Congreso Latino-Ibero-Americano de Investigaciones e Ingenieria de Sistemas, Mexico City, October 1992.

[26] A.K. Chandra, D.S. Hirschberg, C.K. Wong, Approximation algorithms for some generalized knapsack problems, Theoretical Computer Sciences 3 (1976) 293–304.

[27] I. Charon, O. Hudry, The noising method: A new method for combinatorial optimization, Operations Research Letters 14 (1993) 133–137.

[28] P. Chu, J. Beasley, A genetic algorithm for the multidimensional knapsack problem, Journal of Heuristics 4 (1998) 63–86.

[29] CPLEX Optimization, Inc. (1994), Using the CPLEX callable library and CPLEX mixed integer library, Version 3.0.

[30] Y. Crama, J. Mazzola, On the strength of relaxations of multidimensional knapsack problems, INFOR 32 (1994) 219–225.

[31] F. Dammeyer, S. Voss, Application of tabu search strategies for solving multiconstraint zero–one knapsack problems, Working paper, Technische Hochschule Darmstadt, Germany, 1991.

[32] F. Dammeyer, S. Voss, Dynamic tabu list management using reverse elimination method, Annals of Operations Research (1993) 31–46.

[33] DASH Associates, 1994, XPRESS-MP User Manual.

[34] G. Dobson, Worst-case analysis of greedy heuristics for integer programming with non-negative date, Mathematics of Operations Research 7 (1982) 515–531.

[35] A. Drexl, A simulated annealing approach to the multiconstraint zero–one knapsack problem, Computing 40 (1988) 1–8.

[36] G. Dueck, T. Scheuer, Threshold accepting: A general purpose optimization algorithm, Journal of Computational Physics 90 (1990) 161–175.

[37] G. Dueck, J. Wirshing, Threshold accepting algorithms for multi-constraint 0–1 knapsack problems, Working paper, TR 89 10 016, IBM Heidelberg Scientific Center, Germany, 1989.
[38] H.E. Dyer, Calculating surrogate constraints, Mathematical Programming 19 (1980) 255–278.
[39] M.E. Dyer, A.M. Frieze, Probabilistic analysis of the multidimensional knapsack problem, Mathematics of Operations Research 14 (1989) 162–176.
[40] R.E. Echols, L. Cooper, Solution of integer linear programming problems with direct search, Journal of the Association for Computational Machinery 15 (1968) 75–84.
[41] J. Edmonds, Minimum partition of a matroid into independent subsets, Journal of Research of the National Bureau of Standards 69B (1965) 67–72.
[42] J. Etcheberry, C. Conca, E. Stacchetti, An implicit enumeration approach for integer programming using subgradient optimization, Working paper no 78-04-c, Universidad de Chile, 1978.
[43] L.F. Escudero, S. Martello, P. Toth, A framework for tightening 0–1 programs based on extensions of pure 0–1 KP and SS problems, in: E. Balas, J. Clausen (Eds.), Proceedings of the 4th international IPCO Conference, Springer, Berlin, 1996, pp. 110–123.
[44] H. Everett III, Generalized Lagrange multiplier method for solving problems of optimum allocation of resources, Operations Research 11 (1963) 399–417.
[45] B. Faaland, F.S. Hillier, Interior path methods for heuristic integer programming procedures, Operations Research 27 (1979) 1069–1087.
[46] D. Fayard, G. Plateau, Reduction algorithm for single and multiple constraints 0–1 linear programming problems, in: Proceedings of Congress Methods of Mathematical Programming, Zakopane, Poland, 1977.
[47] D. Fayard, G. Plateau, Algorithm 47: An algorithm for the solution of the knapsack problem, Computing 28 (1982) 269–287.
[48] M.L. Fischer, Worst-case analysis of heuristic algorithms, Management Sciences 26 (1980) 1–17.
[49] M.L. Fischer, The Lagrangean relaxation method for solving integer programming problems, Management Sciences 27 (1981) 1–18.
[50] J.F. Fontanari, A statistical analysis of the knapsack problem, Journal of Physics A—Mathematical and General 28 (1995) 4751–4759.
[51] J.F. Fontarani, R. Meir, The statistical mechanics of the Ising perceptron, Journal of Physics A—Mathematical and General 26 (1993) 1077–1089.
[52] A. Fréville, Contribution à l'optimisation en nombres entiers, Habilitation à diriger des recherches, Université de Paris-Nord, France, 1991.
[53] A. Fréville, S. Hanafi, Bornes duales robustes pour le sac à dos bi-dimensionel en variables 0–1, Working paper presented at FRANCORO III, Québec, Canada, May 2001.
[54] A. Fréville, G. Plateau, Méthodes heuristiques performantes pour les programmes en variables 0–1, Working paper, ANO-91, Université de Lille 1, France, 1982.
[55] A. Fréville, G. Plateau, Heuristics and reduction methods for multiple constraints 0–1 linear programming problems, European Journal of Operational Research 24 (1986) 206–215.
[56] A. Fréville, G. Plateau, Hard 0–1 multiknapsack test problems for size reduction methods, Investigacion Operativa 1 (1990) 251–270.
[57] A. Fréville, G. Plateau, Sac-à-dos multidimensionel en variables 0–1: Encadrement de la somme des variables à l'optimum, RAIRO Operations Research 27 (1993) 169–187.
[58] A. Fréville, G. Plateau, An exact search for the solution of the surrogate dual of the 0–1 bidimensional knapsack problem, European Journal of Operational Research 68 (1993) 413–421.
[59] A. Fréville, G. Plateau, An efficient preprocessing procedure for the multidimensional knapsack problem, Discrete Applied Mathematics 49 (1994) 189–212.
[60] A. Fréville, G. Plateau, The 0–1 bidimensional knapsack problem: Towards an efficient high-level primitive tool, Journal of Heuristics 2 (1996) 147–167.
[61] A.M. Frieze, M.R. Clarke, Approximate algorithms for the m-dimensional 0–1 knapsack problem: Worst-case and probabilistic analyses, European Journal of Operational Research 15 (1984) 100–109.
[62] V. Gabrel, A. Knippel, M. Minoux, Exact solutions of multicommodity network optimization problems with general step cost functions, Operations Research Letters 25 (1999) 15–23.
[63] B. Gavish, H. Pirkul, Allocation of databases and processors in a distributed data processing, in: J. Akola (Ed.), Management of Distributed Data Processing, North-Holland, Amsterdam, 1982, pp. 215–231.
[64] B. Gavish, H. Pirkul, Efficient algorithms for solving multiconstraint zero–one knapsack problems to optimality, Mathematical Programming 31 (1985) 78–105.
[65] C.E. Gearing, W.W. Swart, T. Var, Determining the optimal investment policy for the tourism sector of a developing country, Management Sciences 20 (1973) 487–497.
[66] G.V. Gens, E.V. Levner, Complexity and approximation algorithms for combinatorial problems: A survey, Central Economic and Mathematical Institute, Academy of Sciences of the USSR, Moscow, 1979.
[67] S. van de Gerr, L. Stougie, On rates of convergence and asymptotic normality in the multiknapsack problem, Mathematical Programming 51 (1991) 349–358.
[68] A.M. Geoffrion, An improved implicit enumeration approach for integer programming, Operations Research 17 (1969) 437–454.
[69] A.M. Geoffrion, A.B. Nelson, Users instructions for (0–1) integer linear programming code RIP30C", Rand Memorandum RM-5627-PR, 1968.

[70] P.C. Gilmore, R.E. Gomory, The theory and computation of knapsack functions, Operations Research 14 (1966) 1045–1075.
[71] F. Glover, A multiphase-dual algorithm for the zero–one integer programming problem, Operations Research 13 (1965) 879–919.
[72] F. Glover, Surrogate constraints, Operations Research 16 (1968) 741–749.
[73] F. Glover, Surrogate constraints duality in mathematical programming, Operations Research 23 (1975) 434–451.
[74] F. Glover, Heuristics for integer programming using surrogate constraints, Decision Sciences 8 (1977) 156–166.
[75] F. Glover, Tabu search, Part 1, ORSA Journal on Computing 1 (1989) 190–206.
[76] F. Glover, Tutorial on surrogate constraint approaches for optimization in graphs, Working paper, Hearin Center for Enterprise Science, University of Mississippi, USA, 2001.
[77] F. Glover, G.A. Kochenberger, Critical event tabu search for multidimensional knapsack problems, in: I.H. Osman, J.P. Kelly (Eds.), Meta-heuristics: Theory and Applications, Kluwer Academic Publishers, 1996, pp. 407–427.
[78] F. Glover, H. Sherali, Y. Lee, Generating cuts from surrogate constraint analysis for zero–one and multiple choice programming, Computational Optimization and Applications 8 (1977) 154–184.
[79] J. Gotlieb, On the effectivity of evolutionary algorithms for multidimensional knapsack problem, Proceedings of the 4th European Conference of Artificial Evolution, Dunkerque, France, LNCS no. 1829, 1999, pp. 23–27.
[80] C.J. Green, Two algorithms for solving independent multidimensional knapsack problems, Management Sciences Report, no. 110, Carnegie institute of Technology, Graduate School of Industrial Administration, Pittsburgh, USA, 1967.
[81] H. Greenberg, W. Pierskalla, Surrogate mathematical programs, Operations Research 18 (1970) 924–939.
[82] M. Guignard, G. Plateau, G. Yu, An application of Lagrangean decomposition to the 0–1 biknapsack problem, Working paper, Wharton School, University of Pennsylvania, presented at TIMS/ORSA Meeting, Vancouver, Canada, 1989.
[83] M. Guignard, K. Spielberg, Search techniques with adaptive features for certain integer and mixed integer programming problems, in: Proceedings Information Processing 1968, North-Holland Publishing Company, Edinburgh, 1969, pp. 238–244.
[84] M. Guignard, K. Spielberg, Logical reduction methods in zero–one programming (minimal preferred variables), Operations Research 29 (1981) 49–74.
[85] S. Hanafi, Contribution à la résolution de problèmes duaux de grande taille en optimisation combinatoire, PhD thesis, University of Valenciennes, France, 1993.
[86] S. Hanafi, A. Fréville, An efficient tabu search approach for the 0–1 multidimensional knapsack problem, European Journal of Operational Research 106 (1998) 659–675.
[87] S. Hanafi, A. Fréville, Extension of reverse elimination method through a dynamic management of the tabu list, RAIRO Operations Research 35 (2001) 251–267.
[88] S. Hanafi, A. Fréville, A. El Abdellaoui, Comparison of heuristics for the 0–1 multidimensional knapsack problem, in: I.H. Osman, J.P. Kelly (Eds.), Meta-Heuristics: Theory and Applications, Kluwer Academic Publishers, 1996, pp. 449–466.
[89] C. Haul, S. Voss, Using surrogate constraints in genetic algorithms for solving multidimensional knapsack problems, in: D.L. Woodruff (Ed.), Advances in Computational and Stochastic Optimization, Logic Programming, and Heuristic Search, Kluwer Academic Publishers, 1998, pp. 235–251.
[90] M. Held, R.M. Karp, The traveling salesman problem and minimum spanning trees, Operations Research 18 (1970) 1138–1162.
[91] M. Held, R.M. Karp, The traveling salesman problem and minimum spanning trees: Part II, Mathematical Programming 1 (1971) 6–25.
[92] R. Hill, C. Reilly, The effects of coefficient correlation structure in two-dimensional knapsack problems on solution procedure performance, Management Sciences 46 (2000) 302–317.
[93] F.S. Hillier, Efficient heuristic procedures for integer linear programming with an interior, Operations Research 17 (1969) 600–637.
[94] T. Ibaraki, Enumerative approaches to combinatorial optimization—Part II, Annals of Operations Research 11 (1987).
[95] T. Ibaraki, E. Ohashi, H. Mine, A heuristic algorithm for mixed-integer programming problems, Mathematical Programming Study 2 (1974) 115–136.
[96] O.H. Ibarra, C.E. Kim, Fast approximation algorithms for the knapsack and sum of subset problems, Journal of Association for Computing Machinery 22 (1975) 463–468.
[97] IBM Corporation, Optimization subroutine library, Guide and Reference, 1990.
[98] S. Isaka, Double relaxation dynamic programming methods for the multi-dimensional knapsack problem, Master Thesis, Department of Applied Mathematics and Physics, Kyoto University, Japan, 1983.
[99] R.G. Jeroslow, T.H.C. Smith, Experimental results on Hillier's search, Mathematical Programming 9 (1975) 371–376.
[100] E.J. Johnson, G.L. Nemhauser, M.W.P. Savelsbergh, Progress in linear programming-based algorithms for integer programming: An exposition, INFORMS Journal on Computing 12 (2000) 2–23.
[101] S. Kaplan, Solution of the Lorie-Savage and similar integer programming problem by the generalized lagrange multiplier method, Operations Research 14 (1966) 1130–1136.
[102] M.H. Karwan, R.L. Rardin, Some relationships between Lagrangian and surrogate duality in integer programming, Mathematical Programming 17 (1979) 320–334.

[103] M.H. Karwan, R.L. Rardin, Searchability of the composite and multiple surrogate dual functions, Operations Research 28 (1980) 1251–1257.

[104] M.H. Karwan, R.L. Rardin, Surrogate dual multiplier search procedures in integer programming, Operations Research 32 (1984) 52–69.

[105] M.H. Karwan, R.L. Rardin, S. Sarin, A new surrogate dual multiplier search procedure, Naval Research Logistics 34 (1987) 431–450.

[106] S.T. Khuri, A. Batarekh, Heuristics for the integer knapsack problem, in: Proceedings of the 10th International Computer Science Conference, Santiago, Chile, 1990, pp. 161–172.

[107] S. Khuri, T. Back, J. Heitkotter, The zero/one multiple knapsack problem and genetic algorithms, in: Proceedings of the 1994 ACM Symposium on Applied Computing, (SAC'99), ACM Press, London, 1994, pp. 188–193.

[108] D.N. Kleinmuntz, C.E. Kleinmuntz, Multiobjective capital budgeting in not-for-profit hospitals and healthcare systems, Working paper, University of Illinois, Urbana-Champaign, USA, October 2001.

[109] G. Kochenberger, G. McCarl, F. Wymann, A heuristic for general integer programming, Decision Sciences 5 (1974) 36–44.

[110] B. Korte, R. Schnader, On the existence of fast approximate schemes, in: O.L. Mangasarian, R.R. Meyer, S.M. Robinson (Eds.), Nonlinear Programming, Academic Press, 1980, pp. 415–437.

[111] E. Korutcheva, M. Opper, B. Lopez, Statistical mechanics of the knapsack problem, Journal of Physics A: Mathematics and General 27 (1994) L645–L650.

[112] W. Kwak, Y. Shi, H. Lee, C.F. Lee, Capital budgeting with multiple criteria and multiple decision makers, Review of Quantitative Finance and Accounting 7 (1996) 97–112.

[113] M. Laguna, J.P. Kelly, J.L. Gonzalez Velarde, F. Glover, Tabu search for the multilevel generalized assignment problem, European Journal of Operational Research 82 (1995) 176–189.

[114] E.L. Lawler, Fast approximation algorithms for knapsack problems, Mathematics of Operations Research 4 (1979) 339–356.

[115] J.S. Lee, M. Guignard, An approximate algorithm for multidimensional zero–one knapsack problems: A parametric approach, Management Sciences 34 (1988) 402–410.

[116] C.E. Lemke, K. Spielberg, Direct search algorithms for zero–one and mixed-integer programming, Operations Research 15 (1967) 892–914.

[117] A. Lokketangen, F. Glover, Probabilistic move selection in tabu search for zero–one mixed integer programming problems, in: I.H. Osman, J.P. Kelly (Eds.), Meta-Heuristics: Theory and Applications, Kluwer Academic Publishers, 1996, pp. 467–487.

[118] A. Lokketangen, F. Glover, Solving zero–one mixed integer programming problems using tabu search, Special Issue on Tabu Search, European Journal of Operational Research 106 (1998) 624–658.

[119] A. Lokketangen, K. Jornsten, S. Storoy, Tabu search within a Pivot and Complement framework, International Transactions of Operations Research 1 (1994) 305–316.

[120] J. Lorie, L.J. Savage, Three problems in capital rationing, Journal of Business 28 (1955) 229–239.

[121] R. Loulou, E. Michaelides, New greedy-like heuristics for the multidimensional 0–1 knapsack problem, Operations Research 27 (1979) 1101–1114.

[122] M.J. Magazine, M.S. Chern, A note on approximation schemes for multidimensional knapsack problems, Mathematics of Operations Research 9 (1984) 244–247.

[123] M.J. Magazine, O. Oguz, A fully polynomial approximation algorithm for the 0–1 knapsack problem, European Journal of Operational Research 8 (1981) 270–273.

[124] M.J. Magazine, O. Oguz, A heuristic algorithm for the multidimensional zero–one knapsack problem, European Journal of Operational Research 16 (1984) 319–326.

[125] J.W. Mamer, K.E. Schilling, On the growth of random knapsacks, Discrete Applied Mathematics 28 (1990) 223–230.

[126] A.S. Manne, H.M. Markowitz, On the solution of discrete programming problems, Econometrica 25 (1957) 84–110.

[127] B. Mans, Contribution à l'algorithmique non numérique parallèle: Parallélisation de méthodes de recherche arborescente, Thèse de Doctorat, Université Paris 6, France, 1992.

[128] R.E. Marsten and T.L. Morin, MMPD, a computer code for solving multiconstraint knapsack problems in integer variables, Working paper, Operations Research Center, MIT, Cambridge, USA, 1976.

[129] R.E. Marsten, T.L. Morin, Optimal solutions found for Senju and Toyoda'0/1 integer programming problems, Management Sciences 23 (1977) 1364–1365.

[130] R.E. Marsten, T.L. Morin, A hybrid approach to discrete mathematical programming, Mathematical Programming 14 (1978) 21–40.

[131] S. Martello, P. Toth, A new algorithm for the 0–1 knapsack problem, Management Sciences 34 (1988) 633–644.

[132] S. Martello, P. Toth, Knapsack problems: Algorithms and computer implementations, in: Series in Discrete Mathematics and Optimization, Wiley Inter Sciences, New York, 1990.

[133] S. Martello, P. Toth, Upper bounds and algorithms for hard 0–1 knapsack problems, Operations Research 45 (1997) 768–778.

[134] S. Martello, D. Pisinger, P. Toth, New trends in exact algorithms for the 0–1 knapsack problem, European Journal of Operational Research 123 (1999) 325–336.

[135] C. McMillan, D.R. Plaine, Resource allocation via 0–1 programming, Decision Sciences 4 (1973) 119–132.

[136] M. Meanti, A.H.G. Rinnooy Kan, L. Stougie, C. Vercellis, A probabilistic analysis of the multiknapsack value function, Mathematical Programming 46 (1990) 237–247.

[137] H. Meier, N. Christofides, G. Salkin, Capital budgeting under uncertainty—an integrated approach using contingent claims analysis and integer programming, Operations Research 49 (2001) 196–206.

[138] M. Nediak and J. Eckstein, Pivot, cut and dive: A heuristic for 0–1 mixed integer programming, Rutcor research report, Rutgers University, USA, October 2001.

[139] G.L. Nemhauser, M.W.P. Savelsbergh, G.C. Sigismondi, MINTO, a Mixed INTeger optimizer, Operations Research Letters 15 (1993) 47–58.

[140] G.L. Nemhauser, Z. Ullmann, Discrete dynamic programming and capital allocation, Management Sciences 15 (1969) 494–505.

[141] S. Niar, A. Fréville, A parallel tabu search algorithm for the 0–1 multidimensional knapsack problem, International Parallel Processing Symposium'97 IEEE Computer Society Press, Suisse, 1997, pp. 512–516.

[142] M. Ohlssen, C. Peterson, B. Soderberg, Neural networks for optimization problems with inequality constraints: The knapsack problem, Neural Computation 5 (1993) 331–339.

[143] C. Oliva, P. Michelon, C. Artigues, Integrating linear programming and constraint programming techniques for solving mixed-integer programming problems, Working paper, University of Avignon, presented at INFORMS 2001 Annual Meeting, Miami Beach, USA, 4–7 November 2001.

[144] M.A. Osorio, F. Glover, P. Hammer, Cutting and surrogate constraint analysis for improved multidimensional knapsack solutions, Research report, University of Puebla, Mexico, presented at IFORS, Mexico, September 2000, Annals of Operational Research, submitted for publication.

[145] M. Padberg, $(1,k)$-configurations and facets for packing problems, Mathematical Programming 18 (1980) 94–99.

[146] C.C. Petersen, Computational experience with variants of the Balas algorithm applied to the selection of R&D projects, Management Sciences 13 (1967) 736–750.

[147] C.C. Petersen, A capital budgeting heuristic algorithm using exchange operations, AIIE Transactions 6 (1974) 143–150.

[148] H. Pirkul, A heuristic solution procedure for the multiconstraint zero–one knapsack problem, Naval Research Logistics 34 (1987) 161–172.

[149] D. Pisinger, An expanding-core algorithm for the exact 0–1 knapsack problem, European Journal of Operational Research 87 (1995) 175–187.

[150] D. Pisinger, A minimal algorithm for the bounded knapsack problem, ORSA Journal on Computing 12 (2000) 75–84.

[151] A. Plateau, D. Tachat, P. Tolla, A hybrid search combining interior point methods and metaheuristics for 0–1 programming, International Transactions in Operational Research 9 (2002) 731–746.

[152] G. Plateau, Réduction de la taille des problèmes linéaires en variables 0–1, Working paper, ANO-71, Université de Lille 1, France, 1976.

[153] G. Plateau, M. Elkihel, A hybrid method for the 0–1 knapsack problem, Methods of Operations Research 49 (1985) 277–293.

[154] G. Plateau, C. Roucairol, A supercomputer algorithm for the 0–1 multiknapsack problem, in: R. Sharda, B. Golden, E. Wasil, O. Balei, W. Stewart (Eds.), Impacts of Recent Computer Advances on Operations Research, Operations Research Series, 1989, pp. 144–157.

[155] C.J. Pursglove, T.B. Boffey, Heuristic improvement methods: How should starting solutions be chosen, Mathematical Programming Study 13 (1980) 135–142.

[156] G.R. Raidl, An improved genetic algorithm for the multiconstraint knapsack problem, in: Proceedings of the 5th IEEE International Conference on Evolutionary Computation, 1998, pp. 207–211.

[157] C.H. Reilly, Optimization test problems with uniformly distributed coefficients, in: B.L. Nelson, W.D. Kelton, G.M. Clark (Eds.), Proceedings of the 1991 Winter Simulation Conference, Institute of Electrical and Electronics Engineers, Phoenix, USA, 1991, pp. 866–874.

[158] A.H.G. Rinnooy Kan, L. Stougie, C. Vercellis, A class of generalized greedy algorithms for the multi-knapsack problem, Discrete Applied Mathematics 42 (1993) 279–290.

[159] R.H. Roth, An approach to solving linear discrete optimization problems, Journal of the Association for Computing Machinery 17 (1970) 303–313.

[160] M.W.P. Savelsbergh, Preprocessing and probing techniques for mixed integer programming problems, ORSA Journal of Computing 6 (1994) 445–454.

[161] K.E. Schilling, The growth of m-constraint random knapsacks, European Journal of Operational Research 46 (1990) 109–112.

[162] K.E. Schilling, Random knapsacks with many constraints, Discrete Applied Mathematics 48 (1994) 163–174.

[163] S. Senju, Y. Toyada, An approach to linear programming problems with 0–1 variables, Management Sciences 15 (1968) 196–207.

[164] S. Shani, Approximate algorithms for the 0–1 knapsack problem, Journal of Association for Computing Machinery 22 (1975) 115–124.

[165] W. Shih, A branch and bound method for the multiconstraint zero–one knapsack problem, Journal of the Operations Research Society 30 (1979) 369–378.

[166] J.A. Sikorski, A branch-and-bound solver for multiconstraint 0–1 knapsack problems, Working paper, Systems Research Institute, Polish Academy of Sciences, Warszawa, Poland, 1994.

[167] A.L. Soyster, B. Lev, W. Slivka, Zero–one programming with many variables and few constraints, European Journal of Operational Research 2 (1978) 195–201.

[168] K. Spielberg, Algorithms for the simple plant location problem with some side conditions, Operations Research 17 (1969) 85–111.

[169] K. Spielberg, Plant location with generalized search origin, Management Science 19 (1969) 165–168.

[170] A. Straszak, M. Libura, J. Siborski, D. Wagner, Computer-assisted constrained approval voting, Group Decision and Negotiation 2 (1993) 375–385.

[171] K. Szkatula, The growth of multi-constraint random knapsacks with various right-hand sides of the constraints, European Journal of Operational Research 73 (1994) 199–204.

[172] K. Szkatula, The growth of multi-constraint random knapsacks with large right-hand sides of the constraints, Operations Research Letters 21 (1997) 25–30.

[173] K. Szkatula, How fast the random knapsacks with many constraints grow? Working paper presented at ECCO XIII conference, Capri, Italy, 2000.

[174] A. Thesen, A recursive branch and bound algorithm for the multidimensional knapsack problem, Naval Research Quarterly 22 (1975) 341–353.

[175] J. Thiel, S. Voss, Some experiences on solving multiconstraint zero–one knapsack problems with genetic algorithms, INFOR 32 (1994) 226–242.

[176] M. Toulouse, T.G. Crainic, M. Gendreau, Communication issues in designing cooperative multithread parallel searches, in: I.H. Osman, J.P. Kelly (Eds.), Meta-Heuristics: Theory and applications, Kluwer Academic Publishers, 1996, pp. 503–522.

[177] Y. Toyoda, A simplified algorithm for obtaining approximate solutions to zero–one programming problems, Management Sciences 21 (1975) 1417–1427.

[178] C.A. Trauth, R.E. Woolsey, Integer linear programming: A study in computational effectiveness, Management Sciences 15 (1969) 481–494.

[179] M. Vasquez, J.K. Hao, A logic-constrained knapsack formulation and a tabu algorithm for the daily photograph scheduling of an earth observation satellite, Computational Optimization and Applications 20 (2001) 137–157.

[180] M. Vasquez, J.K. Hao, Une approche hybride pour le sac à dos multidimensionel en variables 0–1, RAIRO Operations Research 35 (2001) 415–438.

[181] A. Volgenant, J.A. Zoon, An improved heuristic for multidimensional 0–1 knapsack problems, Journal of the Operational Research Society 41 (1990) 963–970.

[182] S. Voss, Tabu search: Applications and prospects, in: D.Z. Du, P.M. Pardalos (Eds.), Network Optimization Problems, 1993, pp. 333–353.

[183] H.M. Weingartner, Mathematical Programming and the Analysis of Capital Budgeting Problems, Prentice-Hall, Englewoods Cliffs, NJ, 1963.

[184] H.M. Weingartner, D.N. Ness, Method for the solution for the multidimensional 0–1 knapsack problem, Operations Research 15 (1967) 83–103.

[185] F.P. Wyman, Binary programming: A decision rule for selecting optimal vs heuristic techniques, The Computer Journal 16 (1973) 135–140.

[186] L.A. Wolsey, Heuristic analysis, linear programming and branch and bound, Mathematical Programming Study 13 (1980) 121–134.

[187] J. Yang, A computational study on 0–1 knapsack problems generated under explicit correlation induction, MS Thesis, Department of Industrial and Systems Engineering, The Ohio State University, Columbus, USA, 1994.

[188] S.H. Zanakis, Heuristic 0–1 linear programming: An experimental comparison of three methods, Management Sciences 24 (1977) 91–104.