Computers & Operations Research 39 (2012) 19–26

Contents lists available at ScienceDirect

Computers & Operations Research

journal homepage: www.elsevier.com/locate/caor

# Problem reduction heuristic for the 0–1 multidimensional knapsack problem

Raymond R. Hill [a,*], Yong Kun Cho [b], James T. Moore [a]

[a] Air Force Institute of Technology, Department of Operational Sciences, 2950 Hobson Way, Wright-Patterson AFB, OH 45433, USA
[b] 22 Itaewan-Ro, Yongsan-GU, Ministry of National Defense, Seoul, South Korea

ARTICLE INFO

Available online 7 July 2010

*Keywords:*
Heuristic optimization
Design of algorithms
Multi-dimensional knapsack problem
Core problem

ABSTRACT

This paper introduces new problem-size reduction heuristics for the multidimensional knapsack problem. These heuristics are based on solving a relaxed version of the problem, using the dual variables to formulate a Lagrangian relaxation of the original problem, and then solving an estimated core problem to achieve a heuristic solution to the original problem. We demonstrate the performance of these heuristics as compared to legacy heuristics and two other problem reduction heuristics for the multi-dimensional knapsack problem. We discuss problems with existing test problems and discuss the use of an improved test problem generation approach. We use a competitive test to highlight the performance of our heuristics versus the legacy heuristic approaches. We also introduce the concept of computational versus competitive problem test data sets as a means to focus the empirical analysis of heuristic performance.

Published by Elsevier Ltd.

## 1. Introduction

As real systems continue to grow in complexity, optimization problem formulations will continue to get larger and more complex. In these cases, finding exact solutions often requires excessive computing time and computational resources. Further, since these large problem formulations often involve parameters that are just estimates, exact optimal solutions may not have much practical value. In such cases, using an optimization heuristic and quickly obtaining a near-optimal solution may better satisfy a real world practitioner. As a result, heuristics and meta-heuristics continue to garner research interest and provide solutions to actual problems.

The integer knapsack problem (KP), and its generalization, the multidimensional knapsack problem (MKP), are frequently used to model various decision-making processes: manufacturing in-sourcing [1], asset-backed securitization [2], combinatorial auctions [3,4], computer systems design [5], resource-allocation [6], set packing [7], cargo loading [8], project selection [9], cutting stock [10], and capital budgeting (early examples include Lorie and Savage [11], Manne and Markowitz [12], Weingartner [13]). Our focus is on the MKP.

The MKP has the following form:

Maximize

$$Z = \sum_{j=1}^{n} c_j x_j \tag{1}$$

subject to

$$\sum_{j=1}^{n} a_{ij} x_j \le b_i, \quad i = 1, \ldots, m \tag{2}$$

$$x_j \in \{0,1\}, \quad j = 1, \ldots, n \tag{3}$$

where $c_j > 0, b_i > 0$, all $a_{ij} \ge 0$ and $x_j = 1$ if an item is selected, $x_j = 0$ if an item is not selected. Additionally, we require that at least one $a_{ij} > 0$ for each $j$.

## 2. Legacy greedy heuristics

Hooker [14] suggests that the performance of algorithms may be analyzed in two ways: one is to analyze performance analytically relying on the methods of deductive mathematics, and the other is to analyze performance empirically using computational experiments. We employ the empirical approach in this research and use this empirical approach to specifically compare our heuristics to comparable legacy heuristics. We do not consider the performance of meta-heuristics such as tabu search, genetic algorithms, ant colony algorithms, etc.

There are a number of effective greedy solution procedures for the MKP; for instance Toyoda [15], labeled as TOYODA, Senju and Toyoda [16], labeled as S–T, Loulou and Michaelides [17], labeled as L–M, Kochenberger et al. [18], labeled as KOCHEN, and Fox and Scudder [7], labeled as FOX. Our testing includes our implementation of each of these approaches. We also compare results to three recent approaches. Akcay et al. [19] introduced a primal effective capacity heuristic (PECH). Designed for the general MKP, PECH

* Corresponding author. Tel.: +1 937 902 5116; fax: +1 937 656 4943.
*E-mail address:* rayrhill@gmail.com (R.R. Hill).

0305-0548/$ - see front matter Published by Elsevier Ltd.
doi:10.1016/j.cor.2010.06.009

selects items for the knapsack based on capacity and reward and is applied easily to the 0–1 MKP. Bertsimas and Demir [20] describe an approximate dynamic programming approach for the MKP, which we label (B & D). Their method approximates the optimal value at each stage using a suboptimal method which they refer to as a base heuristic. They use an adaptive fixing greedy heuristic as the base heuristic after comparing it to selected legacy greedy approaches. We compare our approaches via results on benchmark test problems.

Problem transformation and problem reduction methods have been used to provide heuristic solutions. For instance, a Lagrangian formulation of a MKP creates an unconstrained problem while a surrogate constraint approach creates a single constraint knapsack problem. Solutions to these transformed problems estimate the optimal solution to the actual problem. Boyer et al. [21] use a surrogate method, with dual variables from the LP-relaxation of (1)–(3) as the surrogate multipliers, and a dynamic programming solution method they label HDP. They also add a limited branch and cut (LBC) improvement phase for a second solution approach, HDP-LBC. We also compare our approaches to the Boyer et al. [21] approaches via results on similar benchmark test problems.

We focus on a problem reduction approach in this work. Problem reduction involves removing variables from the formulation or at least fixing those variables to some pre-determined value. As expounded below, we exploit information from a Lagrangian relaxation formulation of the MKP to dynamically estimate the core problem of the MKP; non-core variables are fixed to values of 0 or 1. We solve this core problem, expand its solution to include the fixed, non-core variables, and return the MKP heuristic solution. We compare our approach to two MKP core-problem approaches, described fully in Section 6.

## 3. Empirical analysis of heuristics

The empirical analysis of heuristics involves the study of heuristic performance over some range of test problems. Such studies should produce performance insights, a ranking among the candidate heuristic performers examined, or both. To gain performance insight, particularly as a function of test problem characteristics, test problem instances should be experimentally designed (see Rardin and Uzsoy [22]). We call such test problems empirical-test-focused problems or simply computational test sets. To gain ranking insight, such as which heuristic is the best, test problems should cover the full range of potential problem instances as realized by fully considering the range of problem characteristics. The random generation of these problems should include systematic control of problem parameters, such as distribution of parameters or correlation structure among the problem parameters. This systematic yet random problem generation ensures the full coverage of defined test problem characteristic ranges (see Cho et al. [23]). We call such test problems ranking-test-focused problems or simply competitive test sets. Results from computational test sets provide performance insight as a function of problem characteristics. Results from competitive test sets provide insights into heuristic performance over the full range of potential problem characteristics and thus improved inferences to heuristic performance on actual problems. In this work, we compare heuristics using all types of problem sets.

Unfortunately, researchers too often limit their use of test problems to those sets that were used by other researchers. These comparative results are fine for benchmarking purposes. However, these legacy test sets generally lack experimental design rigor and do not provide the full range of problem characteristic realizations particularly in terms of constraint tightness (see Cho et al. [23]) and in the correlation levels among sets of problem coefficients (see Table 1 for the Chu and Beasley [24] set as an example). Nearly all legacy test problem sets use similar constraint slackness ratios for each constraint within a particular problem, a practice found in the Chu and Beasley [24] test set and the random instances generated for the analysis in Boyer et al. [21]. Problems are harder to solve, and more realistic, when constraint slackness ratios vary among the constraints [23,25]. We explicitly examine such problem sets in this research.

**Table 1**
Range of objective function to constraint function correlations in Chu and Beasley [24] MKP test problems.

| File | Min $\rho_{CA}$ | Max $\rho_{CA}$ | $(n,m)$ |
|---|---|---|---|
| mknapcb1 | 0.094 | 0.511 | (100, 5) |
| mknapcb2 | 0.163 | 0.461 | (100,10) |
| mknapcb3 | 0.189 | 0.403 | (100, 30) |
| mknapcb4 | −0.157 | 0.459 | (250, 5) |
| mknapcb5 | 0.003 | 0.326 | (250, 10) |
| mknapcb6 | 0.030 | 0.308 | (250, 30) |
| mknapcb7 | −0.256 | 0.437 | (500, 5) |
| mknapcb8 | −0.192 | 0.307 | (500, 10) |
| mknapcb9 | −0.074 | 0.213 | (500, 30) |

$(n,m)$ represents (variables, constraints) in problems.
Each file contains 30 test problems.
The $\rho_{CA}$ is the correlation between the objective function and a constraint.

Interestingly, Kellerer et al. [26] provide a comprehensive review of the family of knapsack problems and include a chapter on the 0–1 MKP. Their Section 5.5 summarizes the common suite of correlation induction strategies: uncorrelated, weakly correlated, strongly correlated, etc. While not quantified, their plots of coefficients from these induction strategies show near perfect correlation for all but the uncorrelated method. Using the analysis method of Reilly [27], Hill and Reilly [28] indicate that these induction schemes induce correlation levels of zero for the uncorrelated method and correlation levels above 0.97 for all other methods. Test problems, either computational or competitive, should cover a full range of test problem coefficient correlation values. See Reilly [29] for a detailed discussion.

## 4. Test sets employed in this research

Hill and Reilly [25] provide a two-dimensional knapsack problem (2KP) test set (1120 problems). Cho et al. [30] developed a five-dimensional knapsack problem (5KP) test set (3780 problems) that varied and controlled problem characteristics, specifically the constraint slackness and correlation structure, across their entire range of values, via a designed experimental approach. For constraint construction, two different constraint slackness values were used. Constraint slackness, $S_i$, is the ratio of the right-hand side value of constraint $i$ to the sum of the coefficients for that constraint; $S_i = b_i / \sum_{j=1}^{n} a_{ij}$. This value is preset and used to generate the right-hand side value in a test problem once the constraint coefficients have been randomly generated. A slackness code of "1" indicates $S_i = 0.30$ (tight constraint) and a slackness code of "2" indicates $S_i = 0.70$ (a loose constraint). Each constraint is generated using its own marginal distribution of coefficients to avoid generating problems with identical constraints.

The study by Hill and Reilly [25] varied constraint slackness settings and problem coefficient correlation structure. The stronger effect on performance was due to constraint slackness setting, but a critical factor in the experimental design employed was the full range of coefficient correlation structures used.

**Table 2**
Comparisons of NR to legacy heuristic performance on computational problems (280 problems per 2KP line, 630 problems per 5KP line).

| | Slackness settings | # times best solution | | | | | |
|---|---|---|---|---|---|---|---|
| | | NR | TOYODA | S–T | L–M | FOX | KOCHEN |
| 2KP | (1, 1) | 110 (170) | 3 (44) | 3 (39) | 5 (13) | 17 (35) | 58 (124) |
| | (1, 2) | 50 (173) | 8 (43) | 23 (155) | 6 (27) | 8 (73) | 19 (140) |
| | (2, 1) | 48 (189) | 9 (55) | 21 (157) | 4 (27) | 2 (79) | 17 (140) |
| | (2, 2) | 72 (205) | 0 (64) | 0 (60) | 8 (35) | 13 (89) | 31 (129) |
| 5KP | (1,1,1,1,1) | 538 (553) | 2 (12) | 1 (7) | 11 (13) | 2 (2) | 54 (70) |
| | (1,1,1,1,2) | 437 (467) | 5 (19) | 4 (12) | 17 (29) | 1 (4) | 120 (154) |
| | (1,1,1,2,2) | 352 (404) | 8 (39) | 9 (30) | 38 (51) | 4 (8) | 144 (202) |
| | (1,1,2,2,2) | 270 (353) | 18 (46) | 31 (101) | 49 (83) | 3 (8) | 131 (229) |
| | (1,2,2,2,2) | 206 (351) | 12 (35) | 71 (217) | 37 (90) | 0 (28) | 97 (221) |
| | (2,2,2,2,2) | 451 (526) | 1 (23) | 0 (20) | 11 (30) | 9 (36) | 72 (125) |

Value in parenthesis is number of times best to include ties
Key:
(#1, #2), Value=1 if tight constraint, value=2 if loose constraint.
NR, New problem reduction heuristic.
TOYODA [15].
S–T, Senju and Toyoda [16].
L–M, Loulou and Michaelides [17].
FOX, Fox and Scudder [7].
KOCHEN, Kochenberger et al. [18].

This full range of coefficient correlation structures removed any confounding effect of correlation on constraint slackness effects. We employ the same experimental design philosophy in the current research.

With five constraints and two levels of constraint tightness, there are 5! =120 constraint combinations. However, many of these combinations show the same mixture of tight and loose constraint settings leaving $2^5$ combinations of potential interest. Since our focus is on the nature of the mixed settings, order among the constraints does not matter; we simply need representative combinations of mixed slackness settings. Thus, to reduce problem test size and still gain the desired insight into heuristic performance on problems with multiple tight constraints, only six combinations are used (see Tables 2 and 3 for the combinations used and note these cover the possible mixture of settings desired). The range of correlation levels within the problems cover the entire feasible correlation range using explicit correlation induction methods [25].

## 5. New heuristic using Lagrange multipliers and the core problem

The Lagrangian relaxation of problem (1)–(3) with an $m$-dimensional vector of Lagrange multipliers is defined as follows:

$$Z_{LR}(\mu) = \max\left\{\sum_{j=1}^{n} c_j x_j + \sum_{i=1}^{m} \mu_i \left(b_i - \sum_{j=1}^{n} a_{ij} x_j\right)\right\} \quad (4)$$

where $x_j \in \{0,1\}$, $j = 1,2,\ldots,n$; $\mu_i \geq 0$, $i = 1,2,\ldots,m$. The solution to problem (4) provides an upper bound on the optimal solution of the original MKP shown in (1)–(3). The best Lagrangian bound is determined by finding multipliers which correspond to

$$Z_{LD}(\mu^*) = \min\{Z_{LR}(\mu) : \mu \geq 0\} \quad (5)$$

This problem is called the Lagrangian dual problem [33]. This upper bound with optimal multipliers $\mu^*$ can equal the bound provided by the LP relaxation of the MKP [34]. However, finding optimal multipliers $\mu^*$ is a difficult combinatorial problem. Dual variables from the solution of the linear programming relaxation of the original MKP are reasonable estimates of the optimal multipliers [31,35]; this research uses values of the dual variables as the multiplier estimates.

Once the multipliers have been found, the Lagrangian relaxation problem, (4), can be solved directly and provide a heuristic solution to the original MKP. For this research we define and use the following efficiency measure:

$$\gamma_j^e = c_j - \sum_{i=1}^{m} \mu_i a_{ij}. \quad (6)$$

If $\gamma_j^e$ is positive, the $j$th variable increases the objective function value. If $\gamma_j^e$ is negative, the $j$th variable decreases the objective function value. Therefore, set $x_j = 1$ if its efficiency measure is positive and $x_j = 0$ if its efficiency measure is non-positive. This solution is the optimal solution of $Z_{LR}(\mu)$ and, if feasible, is a heuristic solution to the original MKP.

Balas and Zemel [36] found that a KP solution obtained by a greedy heuristic using the decreasing order of bang-for-the-buck efficiency ratios (benefit/cost or $c_j/a_j$) differed from the optimal solution among only a few variables. For these few variables, the bang-for-the-buck efficiency measures have insufficient discriminatory power to heuristically provide the optimal solution. Pisinger [37] called this subset of variables the core problem. Pirkul [31] and Pisinger [37] focus on core problem issues for the MKP, in particular, how to estimate the core problem size.

Pirkul [31] defines a core problem for MKP as those values

$$\tau_j = c_j \Big/ \sum_{i=1}^{m} \mu_i a_{ij}$$

that fall "between the maximum and minimum ratios for which $[x_j]$ has a different value in an optimal solution to MKP than that in an optimal solution to a linear programming relaxation." Pirkul's empirical results indicate that the core problem is a small subset of problem variables. However, estimating the core problem is not perfect; we can merely estimate its size.

Pirkul [31] uses this $\tau_j$ value as an efficiency measure, sorts the problem variables in decreasing order by this measure, and fixes each variable to a value of 1, in turn, so long as no constraint is violated (when a violation occurs, the variable is returned to a value of 0). Puchinger et al. [38] varies the efficiency measure used and uses a fixed percentage of problem size as the core problem estimate. Others use a fixed size subset of problem variables to approximate the core. Balas and Zemel [36] used a fixed value of 50, Martello and Toth [39] used $2\sqrt{n}$ for problems of size $n > 200$ and Pisinger [37] used a 100 variable subset as his approximate core. Our approach determines the core problem as a function of the efficiency measure versus using a fixed percentage or fixed size. This approach means the core size will vary dynamically based on problem characteristics.

We found that if $\gamma_j^e$ of (6) has a relatively large positive value, the optimal solution to the original MKP has a tendency for $x_j = 1$ while for a relatively large negative value, $x_j = 0$. The $x_j$ value in the optimal solution is uncertain when $\gamma_j^e$ is near zero. Thus, to ease interpretation, promote problem independence and improve algorithm definition, we normalize the $\gamma_j^e$ to fall in the $[-1,1]$, denote as $\gamma_j$ these normalized efficiency ratios, and place those variables in the core whose $\gamma_j \in [-\alpha,\alpha]$. Non-core variables are fixed to 0 if their $\gamma_j < 0$ and 1 otherwise. The core problem can be solved using a heuristic or exact algorithm.

Let $N=\{x_j, j=1, \ldots, n\}$ be the set of decision variables from (1)–(3). Then define the following index sets:

$$C = \{j | x_j \in N, |\gamma_j| < \alpha, j = 1,\ldots,n\} \quad (7)$$

$$Z = \{j | x_j \in N, j \notin C, \gamma_j < 0, j = 1,\ldots,n\} \quad (8)$$

**Table 3**
Comparison of NR(P) with GLOVER, PIRKUL, NG and NR heuristics in terms of relative error measures.

| | Slackness settings | NR(P) | | PIRKUL | | Glover | NG | NR |
|---|---|---|---|---|---|---|---|---|
| | | Rel. error | Moves | Rel. error | Moves | Rel. error | Rel. error | Rel. error |
| 2KP | (1, 1) | 0.12 | 13.74 | 0.09 | 38.40 | 0.18 | 0.46 | 0.60 |
| | (1, 2) | 0.04 | 13.45 | 0.03 | 38.40 | 0.08 | 0.20 | 0.21 |
| | (2, 1) | 0.03 | 12.67 | 0.02 | 45.77 | 0.07 | 0.15 | 0.16 |
| | (2, 2) | 0.02 | 15.32 | 0.02 | 72.48 | 0.06 | 0.09 | 0.12 |
| 5KP | (1,1,1,1,1) | 0.31 | 14.70 | 0.26 | 33.16 | 0.47 | 0.73 | 0.81 |
| | (1,1,1,1,2) | 0.23 | 13.73 | 0.19 | 36.76 | 0.33 | 0.66 | 0.75 |
| | (1,1,1,2,2) | 0.18 | 12.87 | 0.14 | 40.73 | 0.22 | 0.51 | 0.63 |
| | (1,1,2,2,2) | 0.11 | 11.50 | 0.09 | 45.32 | 0.15 | 0.41 | 0.51 |
| | (1,2,2,2,2) | 0.08 | 10.29 | 0.06 | 51.04 | 0.10 | 0.36 | 0.35 |
| | (2,2,2,2,2) | 0.05 | 16.54 | 0.04 | 67.91 | 0.11 | 0.18 | 0.16 |

Moves represents the number of moves in local improvement phase.
NR(P) is NR with Pirkul improvement; average of 0.02 s/problem.
PIRKUL is from Pirkul [31]; average of 0.07 s/problem.
GLOVER is from Glover [32]; average of 0.12 s/problem.

$$O = \{j | x_j \in N, j \notin C, j \notin Z, j = 1, \ldots, n\} \tag{9}$$

Set $C$ is the set of indices for the variables in the core problem, set $Z$ is the set of indices for the non-core variables fixed to zero and set $O$ is the set of indices for the non-core variables fixed to one. Then define

$$b_i' = b_i - \sum_{j \in O} a_{ij}, \quad i = 1, \ldots, m. \tag{10}$$

The core problem to solve becomes
Maximize

$$Z' = \sum_{j \in C} c_j x_j \tag{11}$$

subject to

$$\sum_{j \in C} a_{ij} x_j \le b_i', \quad i = 1, \ldots m \tag{12}$$

$$x_j \in \{0,1\}, \quad j \in C \tag{13}$$

The core problem solution values, combined with $x_j = 0\ \forall j \in Z$ and $x_j = 1\ \forall j \in O$ yields the heuristic solution to (1)–(3).

Our new reduction approach procedure (NR) is summarized as follows:

*Step* 1: Obtain Lagrange multipliers using the linear programming relaxation of (1)–(3).

*Step* 2: Transform the MKP into an unconstrained Lagrangian problem as in (4).

*Step* 3: Use $\gamma_j$ (normalized from (6)) to categorize the $x_j$ according to (7)–(9) (using $\alpha = 0.15$).

*Step* 4: Solve the approximate core problem, (11)–(13), using a heuristic or exact algorithm and return the final heuristic solution to (1)–(3).

Fig. 1 gives some idea of the dynamic nature of our approach. The figure plots normalized $\gamma_j$ versus variable $i$ for two selected 2KP problems. The figure on the left is for a hard to solve problem; the figure on the right is for an easy to solve problem. Fig. 2 provides the corresponding histograms. The harder problem has more variables with their $\gamma_j$ near zero; the easier problem shows fewer $\gamma_j$ values around the zero mark. Using a simple percentage-based approach or fixed core problem size, as others have employed, yields an approximate core that either underestimates core size or overestimates core size. We contend that the size of the approximate core should vary dynamically based on problem structure and that structure is captured in our efficiency measure used to determine the core. Our approach means a larger core problem size as needed for harder to solve problems, a smaller core problem size for easier to solve problems.

Using the normalized range $\gamma_j \in [-0.15, 0.15]$ estimates a core problem with an average of 27.5 variables in the 2KP test cases and 28.1 variables in the 5KP test cases. Detailed computational efforts, similar to those used in Pirkul [31], found that the range of $\gamma_j \in [-0.15, 0.15]$ includes the actual core problems in 93% of the 2KP problems (1042 of 1120 problems) and in 96% of 5KP problems (3625 of 3780 problems). The range of $\gamma_j$ can of course be expanded to include more variables; however, this increase makes the larger core problem more difficult to solve without a guarantee of optimality. The NG heuristic of Cho et al. [23] is used to solve the core problem. In subsequent comparisons, the NG heuristic is also used independently.

Table 2 compares NR to various legacy approaches using the 2KP and the 5KP computational test sets. Our measure is the number of times each heuristic uniquely found the best solution. The value provided in parentheses is the number of times the heuristic found the best solution to include counting ties for the best solution.

Table 2 results suggest that NR is robust against various constraint slackness and correlation structures as compared to the set of legacy heuristics and outperforms the non-problem reduction heuristics considered. Note KOCHEN is outperformed by S–T on the 2KP problems when slackness settings are mixed (something exploited in Cho et al. [23]).

## 6. New reduction heuristic with an improvement phase

Glover [32] and Pirkul [31] combine a greedy heuristic with a local improvement phase. Their problem reduction approaches are implemented and referred to as GLOVER and PIRKUL, respectively. The PIRKUL improvement phase was added to our NR approach yielding the approach we refer to as NR(P).

An effective local improvement method should (1) use few iterations, or moves, and (2) improve the solution from which it started. Table 3 compares NR(P) performance to GLOVER and PIRKUL. Also included is the NG heuristic [23] and NR.

Table 3 results provide a competitive test, so we report the results using the generally accepted measures of relative error and computational effort. Table 4 reports how often each approach provided the best solution among the considered suite of approaches. NR(P) solution quality results are near the PIRKUL results and better than the GLOVER results. However, NR(P) uses significantly fewer moves (the Moves column) in the local improvement phase due to our improved estimate of the core

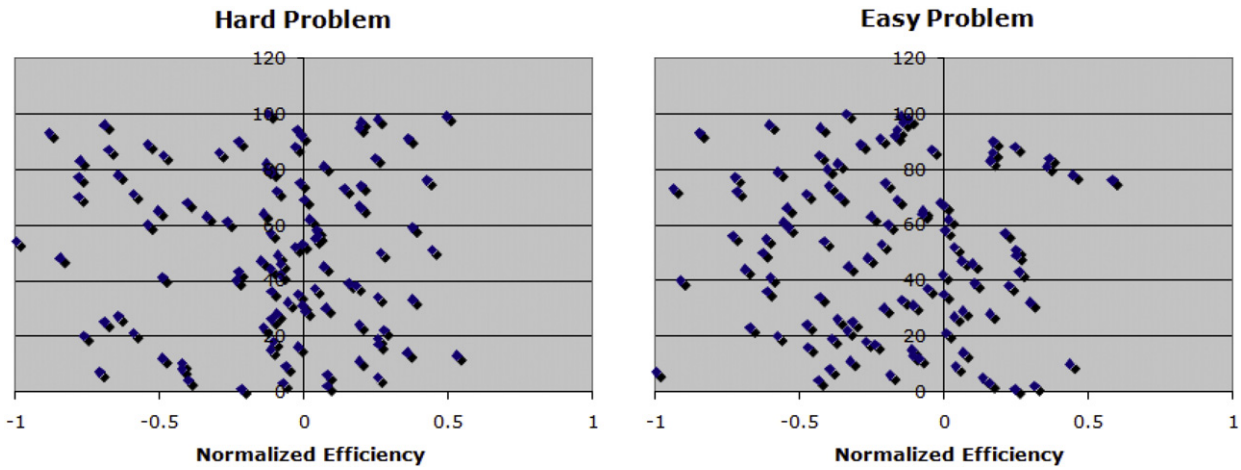


**Fig. 1.** Example density of variables around zero $\gamma_j$, for a hard and an easy problem.

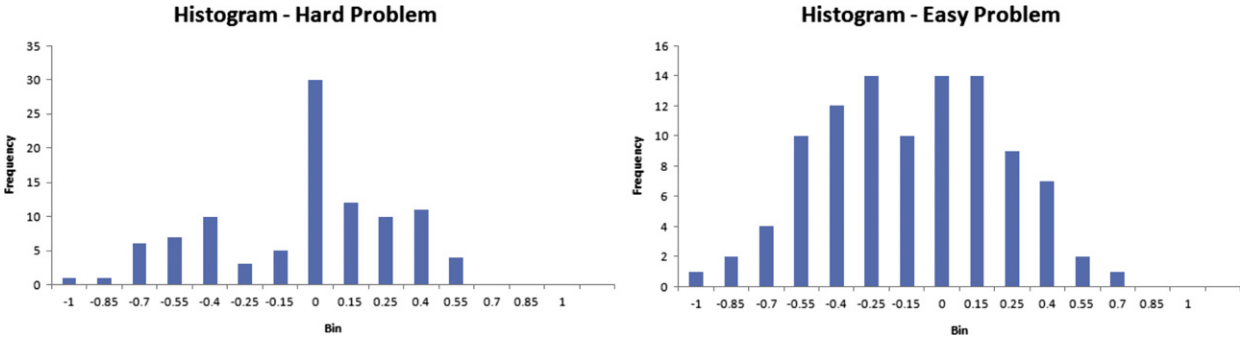


**Fig. 2.** Histograms of $\gamma_j$, for representative hard (left) and an easy (right) problem.

**Table 4**
Comparison of NR(P) with GLOVER, PIRKUL, NG and NR heuristics in terms of number times finding best solution.

| | Slackness | NR(P) | PIRKUL | Glover | NG | NR |
|---|---|---|---|---|---|---|
| 2KP | (1, 1) | 192 | 220 | 138 | 64 | 52 |
| | (1, 2) | 238 | 261 | 173 | 103 | 103 |
| | (2, 1) | 244 | 257 | 174 | 105 | 106 |
| | (2, 2) | 239 | 251 | 155 | 99 | 96 |
| 5KP | (1,1,1,1,1) | 114 | 125 | 75 | 29 | 28 |
| | (1,1,1,1,2) | 163 | 187 | 119 | 38 | 32 |
| | (1,1,1,2,2) | 213 | 245 | 179 | 72 | 63 |
| | (1,1,2,2,2) | 255 | 286 | 228 | 81 | 76 |
| | (1,2,2,2,2) | 327 | 354 | 304 | 109 | 120 |
| | (2,2,2,2,2) | 263 | 305 | 167 | 81 | 97 |

NR(P) is NR with Pirkul improvement.
PIRKUL is from Pirkul [31].
GLOVER is from Glover [32].

problem size. The GLOVER implementation always uses 50 Moves. Average processing times for the heuristics (as applied to the 2KP instances) are shown in the Legend area in the bottom of the table; note all times are in seconds, very comparable, but favor NR(P). Interestingly, NG outperforms NR even though the actual values are very close. This is caused by the bias introduced due to the uncertainty of the core estimate. The local improvement phase added to NR, yielding NR(P), overcomes this bias ultimately providing the best results in the table.

Table 5 compiles comparative results of our NR and NR(P) heuristics against PECH from Akcay et al. [19], the approximate dynamic programming approach of Bertsimas and Demir [20] and the heuristics from Boyer et al. [21], using the Chu and Beasley [24] test set for benchmark purposes. Akcay et al. [19] list the Chu and Beasley [24] genetic algorithm (GA) results, so these are included in our table as well. For comparative purposes, our results are presented in the format used by most of the other authors, breaking problem results out by problem size and $S_i$ value used for each constraint. Each line involves 10 problem instances and the averages are for the 30 instances in each $(m, n)$ combination, which also corresponds to each file. Despite the limited data from Bertsimas and Demir [20], it is clear that our approaches easily better PECH and the approximate dynamic programming approach, B&D. The GA approach of Chu and Beasley [24] betters NR(P) in the majority of the cases (18 of 27) but overall relative error values are quite close. The NR(P) results are most comparable to the GA approach when slack values are high; these problems typically involve more variables set to a value of 1 in the final solution (see Cho et al. [23] for rationale).

The Boyer et al. [21] approaches are the most recent published results but are only averaged by $(m, n)$ combination; the other published works as well as ours drill down into the 10 problem instances within each 30 $(m, n)$ combination. The NR(P) approach is competitive with HDP; neither approach has an advantage over the other. Even with the LBC phase added to HDP, yielding HDP-LBC, NR(P) is comparable, besting HDP-LBC on two problem sets and extremely close on others. Since NR(P) runs in less than 1 s on all problems, NR(P) offers a large advantage in processing times over each of the competitor approaches summarized.

## 7. Analyses on competitive-test focused test problems

Results thus far have involved the use of computational test problems and a comparison using a legacy test problem set for

**Table 5**
Comparison of recent algorithms to both NR and NR(P) based on the Chu and Beasley [24] test set.

| $m$ | $n$ | Slack $S_i$ | Mean relative error | | | | | | $HDP-LBC$[c] |
|---|---|---|---|---|---|---|---|---|---|
| | | | $PECH$[a] | $GA$[a] | $B\&D$[b] | NR | NR(P) | $HDP$[c] | |
| 5 | 100 | 0.25 | 7.34 | 0.99 | n/a | 1.88 | 0.94 | | |
| | | 0.50 | 3.47 | 0.45 | n/a | 1.05 | 0.44 | | |
| | | 0.75 | 2.02 | 0.32 | n/a | 0.59 | 0.22 | | |
| | | Average | 4.28 | 0.59 | | 1.17 | 0.53 | 0.69 | 0.57 |
| 5 | 250 | 0.25 | 7.12 | 0.23 | n/a | 1.04 | 0.46 | | |
| | | 0.50 | 3.20 | 0.12 | n/a | 0.45 | 0.17 | | |
| | | 0.75 | 1.77 | 0.08 | n/a | 0.25 | 0.10 | | |
| | | Average | 4.03 | 0.14 | | 0.58 | 0.24 | 0.22 | 0.16 |
| 5 | 500 | 0.25 | 6.41 | 0.09 | n/a | 0.45 | 0.15 | | |
| | | 0.50 | 3.43 | 0.04 | n/a | 0.21 | 0.06 | | |
| | | 0.75 | 1.70 | 0.03 | n/a | 0.13 | 0.03 | | |
| | | Average | 3.85 | 0.05 | | 0.27 | 0.08 | 0.08 | 0.07 |
| 10 | 100 | 0.25 | 8.20 | 1.56 | n/a | 2.83 | 2.05 | | |
| | | 0.50 | 3.73 | 0.79 | n/a | 1.06 | 0.81 | | |
| | | 0.75 | 1.82 | 0.48 | n/a | 0.62 | 0.44 | | |
| | | Average | 4.58 | 0.94 | | 1.51 | 1.10 | 1.22 | 0.95 |
| 10 | 250 | 0.25 | 5.86 | 0.51 | n/a | 1.17 | 0.88 | | |
| | | 0.50 | 2.56 | 0.25 | n/a | 0.56 | 0.39 | | |
| | | 0.75 | 1.52 | 0.15 | n/a | 0.27 | 0.19 | | |
| | | Average | 3.31 | 0.30 | | 0.67 | 0.48 | 0.46 | 0.32 |
| 10 | 500 | 0.25 | 5.06 | 0.24 | n/a | 0.52 | 0.34 | | |
| | | 0.50 | 2.45 | 0.11 | n/a | 0.26 | 0.14 | | |
| | | 0.75 | 1.23 | 0.07 | n/a | 0.14 | 0.10 | | |
| | | Average | 2.92 | 0.14 | | 0.31 | 0.19 | 0.21 | 0.16 |
| 30 | 100 | 0.25 | 6.83 | 2.91 | n/a | 2.24 | 2.24 | | |
| | | 0.50 | 3.19 | 1.34 | n/a | 1.33 | 1.32 | | |
| | | 0.75 | 1.91 | 0.83 | n/a | 0.80 | 0.80 | | |
| | | Average | 3.98 | 1.69 | | 1.45 | 1.45 | 2.04 | 1.81 |
| 30 | 250 | 0.25 | 4.83 | 1.19 | 1.61 | 1.44 | 1.27 | | |
| | | 0.50 | 2.08 | 0.53 | 0.69 | 0.76 | 0.75 | | |
| | | 0.75 | 1.16 | 0.31 | 0.53 | 0.39 | 0.38 | | |
| | | Average | 2.69 | 0.68 | 0.94 | 0.86 | 0.80 | 0.89 | 0.77 |
| 30 | 500 | 0.25 | 3.69 | 0.61 | 0.81 | 0.92 | 0.89 | | |
| | | 0.50 | 1.70 | 0.26 | 0.43 | 0.37 | 0.36 | | |
| | | 0.75 | 0.88 | 0.17 | 0.29 | 0.24 | 0.23 | | |
| | | Average | 2.09 | 0.35 | 0.51 | 0.51 | 0.49 | 0.48 | 0.42 |

$m$ represents the number of constraints and $n$ represents number of variables.

[a] From Akcay et al. [19]; PECH is their heuristic, GA from Chu and Beasley [24].

[b] From Bertsimas and Demir [20]; data are incomplete and from their best performer.

[c] From Boyer et al. [21].

**Table 6**
Average relative error by heuristic based on the Chu and Beasley [24] test set.

| Test problem | TOYODA | S–T | L–M | FOX | KOCHEN | NR | NR(P) | NG |
|---|---|---|---|---|---|---|---|---|
| mknapcb1 | 2.81 | 3.56 | 5.74 | 9.47 | 0.97 | 1.17 | 0.53 | 0.99 |
| mknapcb2 | 2.09 | 2.46 | 6.58 | 9.82 | 0.44 | 0.58 | 0.24 | 0.41 |
| mknapcb3 | 1.47 | 1.99 | 6.93 | 9.87 | 0.21 | 0.27 | 0.08 | 0.22 |
| mknapcb4 | 3.89 | 4.46 | 4.87 | 11.49 | 1.91 | 1.51 | 1.10 | 1.63 |
| mknapcb5 | 2.81 | 3.38 | 4.35 | 11.26 | 0.81 | 0.67 | 0.48 | 0.61 |
| mknapcb6 | 1.91 | 2.24 | 4.43 | 10.90 | 0.32 | 0.31 | 0.19 | 0.28 |
| mknapcb7 | 4.87 | 5.36 | 3.20 | 12.72 | 2.25 | 1.45 | 1.45 | 1.73 |
| mknapcb8 | 3.74 | 3.90 | 2.87 | 12.76 | 1.39 | 0.86 | 0.80 | 1.01 |
| mknapcb9 | 2.82 | 2.96 | 2.56 | 12.06 | 0.79 | 0.51 | 0.49 | 0.59 |
| Average | 2.93 | 3.37 | 4.61 | 11.15 | 1.00 | 0.81 | 0.60 | 0.83 |

**Table 7**
Number of times best for each heuristic based on Chu and Beasley [24] test set (total best to include ties indicated in parentheses).

| Test problem | TOYODA | S–T | L–M | FOX | KOCHEN | NR | NR(P) | NG |
|---|---|---|---|---|---|---|---|---|
| mknapcb1 | 0 | 0 | 0 | 0 | 2 (4) | 0 (3) | 21 (24) | 4 (6) |
| mknapcb2 | 0 | 0 | 0 | 0 | 3 (4) | 0 (2) | 23 (25) | 2 (2) |
| mknapcb3 | 0 | 0 | 0 | 0 | 0 (0) | 0 (0) | 30 (30) | 0 (0) |
| mknapcb4 | 0 | 0 | 0 | 0 | 3 (3) | 0 (9) | 11 (20) | 7 (9) |
| mknapcb5 | 0 | 0 | 0 | 0 | 3 (3) | 0 (6) | 14 (20) | 7 (7) |
| mknapcb6 | 0 | 0 | 0 | 0 | 2 (2) | 0 (4) | 20 (24) | 4 (4) |
| mknapcb7 | 0 | 0 | 0 | 0 | 3 (3) | 0 (19) | 0 (19) | 8 (12) |
| mknapcb8 | 0 | 0 | 0 | 0 | 2 (2) | 0 (16) | 3 (19) | 9 (10) |
| mknapcb9 | 0 | 0 | 0 | 0 | 1 (1) | 0 (18) | 3 (21) | 8 (8) |

Value in parenthesis is the number of times best to include ties.

**Table 8**
General information pertaining to new MKP test set.

| Parameters | Values |
|---|---|
| Problems | 270 problems, 30 problems per 9 files |
| Variables | 50, 100, 250 |
| Constraints | 5, 10, 25 |
| Slackness | Set $\alpha_i \sim U(0.2,0.8)$ and varied in each constraint |
| Correlation | Define $\rho_{CA^i} \sim U(-0.9,0.9)$ |
| | Set each $\rho_{A^iA^j}$ to ensure positive semi-definiteness |
| Coefficient | Objective function coefficients, $C_j \sim DU(1,100)$ |
| Distribution | Constraint function coefficients, $A_{ij} \sim DU(1,r_j), r_j \sim DU(40,90)$ |

Note: U() indicates uniform distribution, DU() indicates discrete uniform distribution.

benchmarking purposes. We close with a competitive test among the heuristics considered using the legacy data set from Chu and Beasley [24] and the competitive data set from Cho et al. [23]. In this section, we also employ the reduction heuristic with an exact algorithm, branch-and-bound [40], used to solve the core problem. We label this approach NR(X).

Table 6 provides the results using the five legacy greedy heuristics against NR, NR(P), and NG on the Chu and Beasley [24] test set. Table 6 presents solution quality as the percentage of the optimal solution. Two things should be noted. First, all the heuristics perform consistently across the problem sets, even as the size of the problems increase (the left most column provides the file name for each 30-problem subset; problem subsets increase in the number of variables and the number of constraints as listed in Table 5). Second, both NR and NR(P) provide the best performance. Table 7 reports how well each heuristic performs in terms of providing a unique best solution (number of times each heuristic provides the best solution to include ties for the best is indicated in parentheses).

Cho et al. [23] present a competitive MKP test set. Five problem-generation parameters are varied and controlled: number of variables, number of constraints, the constraint slackness, the correlation value, and the coefficient distribution. Nine test files containing 270 problems were created (30 problems in each file). Each file has a different combination of number of variables and number of constraints as follows: 50-5KP, 100-5KP, 250-5KP, 50-10KP, 100-10KP, 250-10KP, 50-25KP, 100-25KP, and 250-25KP (number of variables-number of constraints). Table 8 provides the generation details for the competitive test set.

Table 9 provides the competitive test results. The NR(X) approach involves using a branch and bound (exact) approach to solve the core. Several insights can be gleaned from these results.

Note that as problem size increases, and the problems get more difficult to solve, the performance of the greedy legacy heuristics degrade; this degradation in performance is not observed in Table 6 involving the legacy data set. Note, however, that each of NG [23], NR and NR(P) do not demonstrate a performance degrade. The NR and NR(P) heuristics are more

**Table 9**
Average relative error by each heuristic solving new competitive test set.

| Test problem | Toyoda | S-T | L-M | FOX | KOCHEN | NR | NR(P) | | Pirkul | | NR(X) | | Num. of nodes | NG |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | Qual. | Moves | Qual. | Moves | Qual. | Nodes | | |
| 50-5KP | 4.27 | 5.82 | 2.43 | 7.74 | 2.22 | 1.00 | 0.22 | 6.9 | 0.18 | 21.3 | 0.01 | 60.8 | 182.7 | 0.834 |
| 100-5KP | 5.51 | 6.42 | 3.94 | 6.15 | 1.77 | 0.47 | 0.10 | 16.6 | 0.07 | 43.7 | 0.00 | 166.3 | 259.9 | 0.403 |
| 250-5KP | 5.84 | 5.92 | 6.18 | 5.47 | 0.89 | 0.20 | 0.04 | 36.4 | 0.03 | 106.1 | 0.00 | 798.4 | 989.6 | 0.210 |
| 50-10KP | 6.55 | 6.76 | 3.21 | 12.05 | 3.40 | 2.33 | 0.92 | 6.4 | 0.73 | 17.1 | 0.08 | 95.7 | 209.6 | 1.685 |
| 100-10KP | 7.57 | 7.69 | 3.94 | 10.49 | 2.66 | 0.94 | 0.27 | 13.9 | 0.11 | 36.5 | 0.00 | 396.2 | 694.3 | 0.832 |
| 250-10KP | 10.46 | 10.67 | 7.66 | 10.18 | 2.45 | 0.32 | 0.07 | 37.3 | 0.06 | 89.3 | 0.00 | 1994.4 | 2585.0 | 0.344 |
| 50-25KP | 9.15 | 10.55 | 4.46 | 15.32 | 6.84 | 2.24 | 0.93 | 6.4 | 0.49 | 13.6 | 0.07 | 119.9 | 309.5 | 1.827 |
| 100-25KP | 10.52 | 12.85 | 5.65 | 13.99 | 5.75 | 1.08 | 0.54 | 13.8 | 0.41 | 28.13 | 0.00 | 792.5 | 1288.0 | 1.139 |
| 250-25KP | 13.24 | 14.72 | 8.51 | 13.74 | 5.02 | 0.52 | 0.18 | 35.0 | 0.15 | 70.3 | 0.00 | 6834.5 | 8520.0 | 0.507 |
| Average | 8.12 | 9.04 | 5.11 | 10.57 | 3.45 | 1.01 | 0.36 | 19.2 | 0.25 | 47.3 | 0.02 | 1251.0 | 1671.0 | 0.867 |

Notes: Qual. represents solution quality measured as relative error.
Moves represents number of moves in core improvement phase.
Nodes represents branch-and-bound nodes employed to find solution.

robust to varied problem structures. Among the greedy approaches, NR(P) and NG are the best.

A comparison between NR and NR(P) results depict the utility of a local improvement phase to overcome the bias introduced due to the uncertainty of the core problem estimate. In all cases, NR(P) attained a significant improvement over the NR solution.

The PIRKUL approach does very well when compared to our NR(P). Relative error results are very close, 0.12% on average in favor of PIRKUL but the NR(P) uses fewer local improvement moves (19.2 on average compared to 47.3).

The NR(X) column yields the best result; in this case we solved the core problem exactly. These results attest to the utility of estimating the core to reduce overall exact processing time. The Nodes column gives the branch-and-bound nodes required to solve the core; the Number of Nodes column gives the branch-and-bound nodes required to exactly solve the full problem. On average, reducing to the core problem saves about 25% of the exact solution effort. These results are also very competitive with Puchinger et al. [38] reported results; however, since different test sets are used, definitive comparisons are difficult.

## 8. Conclusion

We introduced a core problem heuristic approach for the MKP that diverges from the standard practice of a problem size percentage or fixed size core problem estimate. Our approaches use an efficiency measure range approach that provides a dynamic aspect to core problem size estimates making them sensitive to the relative difficulty of the problem. The core problem was solved using either a heuristic or exact approach and the resulting heuristic solution to the MKP was compared to legacy greedy heuristics and problem reduction approaches. Overall, our approaches compared well in terms of solution quality and core problem size estimation and showed robust effectiveness as problem difficult increased on problems featuring varied coefficient correlation structures and constraint slackness levels.

We also established a categorization among test problem data sets. Legacy data sets are those sets used in other research efforts, that are useful for benchmark purposes but are not particularly useful for insight-generating experimentation or competitive heuristic testing. Computational problems provide systematic control over specified problem parameters and are useful for assessing heuristic performance as a function of problem characteristics. Finally, competitive test problems provide random instances that collectively expose the examined heuristics to a full range of problem characteristics and thus better represent the actual problem instances possible. This research examined heuristic performance using all three types of problems.

We believe our work establishes the need for further systematic study of heuristic performance as a function of problem structure and the need for better problem generation methods. Future research avenues include examining other classes of problems using our analysis and test problem generation methodology and exploiting new empirical knowledge to uncover better move neighborhoods for both greedy heuristic and meta-heuristic solution approaches. In terms of further investigation of the core problem solution approaches, finding the best $\mu^*$, or investigating sensitivity of values of $\mu$ is one avenue while a second would be examining the choice of $\alpha$ in the core selection process and whether problem structure might dictate a choice of $\alpha$.

## Acknowledgements

Yung Kun Cho and James T. Moore were partially supported by an Air Force Office of Scientific Research Grant FIATA04336J001. Raymond R. Hill was supported by Air Force Office of Scientific Research Grant FA9550-04-1-0163. The test problems generated in this work can be obtained from the first author.

## References

[1] Cherbaka NS, Mueller RD, Ellis KP. Multidimensional knapsack problems and their application to solving manufacturing insourcing problems. In: Proceedings of the annual industrial engineering research conference; 2004.
[2] Mansini R, Speranza M. A multidimensional knapsack model for the asset-backed securitization. Journal of the Operational Research Society 2002;53(8):822–32.
[3] deVries S, Vohra RV. Combinatorial auctions: a survey. INFORMS Journal on Computing 2003;15(3):284–309.
[4] Rothkopf M, Pekec A, Harstad M. Computationally manageable combinatorial auctions. Technical Report, Piscataway, NJ: Rutgers University; 1995.
[5] Ferreira C, Grotschel M, Kiefl S, Krispenz C, Martin A, Weismantel R. Some integer programs arising in the design of mainframe computers. ZOR-Methods Models Operations Research 1993;38(1):77–110.
[6] Johnson E, Kostreva M, Suhl U. Solving 0–1 integer programming problems arising from large scale planning models. Operations Research 1985;33:805–19.
[7] Fox G, Scudder G. A heuristic with tie breaking for certain 0–1 integer programming models. Naval Research Logistics 1985;32(4):613–23.
[8] Shih W. A branch and bound method for the multiconstraint zero-one knapsack problems. Journal of the Operational Research Society 1979;30:369–78.
[9] Peterson C. Computational experience with variants of the balas algorithm applied to the selection of research and development projects. Management Science 1967;13:736–50.
[10] Gilmore P, Gomery RE. The theory and computation of knapsack functions. Operations Research 1966;14:1045–74.

[11] Lorie J, Savage L. Three problems in capital rationing. Journal of Business 1955;28:229–39.
[12] Manne A, Markowitz H. On the solution of discrete programming problems. Econometrica 1957;25:84–110.
[13] Weingartner H. Capital budgeting of interrelated projects: survey and synthesis. Management Science 1966;12:485–516.
[14] Hooker JN. Needed: an empirical science of algorithms. Operations Research 1994;42(2):201–12.
[15] Toyoda Y. A simplified algorithm for obtaining approximate solutions to zero-one programming problems. Management Science 1975;21(12):1417–27.
[16] Senju S, Toyoda Y. An approach to linear programming with 0–1 variables. Management Science 1968;15(4):B196–207.
[17] Loulou R, Michaelides E. New greedy heuristics for the multidimensional 0–1 knapsack problem. Operations Research 1979;27(6):1101–14.
[18] Kochenberger G, McCarl B, Wyman F. A heuristic for general integer programming. Decision Sciences 1974;5(1):36–44.
[19] Akcay Y, Li H, Xu S. Greedy algorithm for the general multidimensional knapsack problem. Annals of Operations Research 2007;150(1):17–29.
[20] Bertsimas D, Demir R. An approximate dynamic programming approach to multidimensional knapsack problems. Management Science 2002;48(4): 550–65.
[21] Boyer V, Elkihel M, El Baz D. Heuristics for the 0–1 multidimensional knapsack problem. European Journal of Operational Research 2009;199(2):658–64.
[22] Rardin R, Uzsoy R. Experimental evaluation of heuristic optimization algorithms: a tutorial. Journal of Heuristics 2001;7:261–304.
[23] Cho Y, Moore JT, Hill RR, Reilly CH. Exploiting empirical knowledge for bi-dimensional knapsack problem heuristics. International Journal of Industrial and Systems Engineering 2008;3(5):530–48.
[24] Chu P, Beasley J. A genetic algorithm for the multidimensional knapsack problem. Journal of Heuristics 1998;4(1):63–86.
[25] Hill RR, Reilly CH. The effects of coefficient correlation structure. Management Science 2000;46(2):302–17.
[26] Kellerer H, Pferschy U, Pisinger D. Knapsack problems. Springer-Verlag; 2004.
[27] Reilly CH. Input models for synthetic optimization problems. In: Farrington PA, Nembhard HB, Sturrock DT, Evans GW, editors. Proceedings of the 1999 winter simulation conference; 1999. p. 116–21.
[28] Hill RR, Reilly CH. Multivariate composite distributions for coefficients in synthetic optimization problems. European Journal of Operational Research 2000;121(1):64–77.
[29] Reilly CH. Synthetic optimization problem generation: show us the correlations. INFORMS Journal on Computing 2009;21(3):458–67.
[30] Cho Y, Moore JT, Hill R. Developing a new greedy heuristic based on knowledge gained via structured empirical testing. International Journal of Industrial Engineering 2003;4(1):504–10.
[31] Pirkul H. A heuristic solution procedure for the multiconstraint zero-one knapsack problem. Naval Research Logistics 1987;34(2):161–72.
[32] Glover F. Heuristics for integer programming using surrogate constraints. Decision Sciences 1977;8(1):156–66.
[33] Nemhauser G, Wolsey L. Integer and combinatorial optimization. New York: John Wiley and Sons, Inc; 1988.
[34] Fréville A, Hanafi S. The multidimensional 0–1 knapsack problem-bounds and computational aspects. Annals of Operations Research 2005;139(1): 195–227.
[35] Fisher M. The lagrangian relaxation method for solving integer programming problems. Management Science 1981;27(1):1–18.
[36] Balas E, Zemel E. An algorithm for large zero-one knapsack problems. Operations Research 1980;28(5):1130–54.
[37] Pisinger D. Core problems in knapsack algorithms. Operations Research 1999;47(4):570–5.
[38] Puchinger J, Raidl GR, Perschy U. The core concept for the multidimensional knapsack problem. In: Evolutionary computation in combinatorial optimization—EvoCOP 2006. Springer; 2006. p. 195–208.
[39] Martello S, Toth P. A new algorithm for the 0–1 knapsack problem. Management Science 1988;34(5):633–44.
[40] Thesen A. A recursive branch and bound algorithm for the multidimensional knapsack problem. Naval Research Logistics 1975;34(2):341–53.