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.

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 ρ_{CA}	Max ρ_{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 ρ_{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.