European Journal of Operational Research 258 (2017) 866–876

Contents lists available at ScienceDirect

European Journal of Operational Research

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

Discrete Optimization

Tolerance analysis for 0–1 knapsack problems

David Pisinger[a,*], Alima Saidi[b]

[a] DTU Management Engineering, Technical University of Denmark, Produktionstorvet 424, DK-2800 Kgs. Lyngby, Denmark
[b] DIKU, Dept. of Computer Science, University of Copenhagen, Universitetsparken 5, DK-2100 Copenhagen O, Denmark

ARTICLE INFO

Article history:
Received 17 August 2016
Accepted 31 October 2016
Available online 5 November 2016

Keywords:
Robustness & sensitivity analysis
Knapsack problem
Post-optimal analysis
Dynamic programming

ABSTRACT

Post-optimal analysis is the task of understanding the behavior of the solution of a problem due to changes in the data. Frequently, post-optimal analysis is as important as obtaining the optimal solution itself. Post-optimal analysis for linear programming problems is well established and widely used. However, for integer programming problems the task is much more computationally demanding, and various approaches based on branch-and-bound or cutting planes have been presented. In the present paper, we study how much coefficients in the original problem can vary without changing the optimal solution vector, the so-called tolerance analysis. We show how to perform exact tolerance analysis for the 0–1 knapsack problem with integer coefficients in amortized time $O(c \log n)$ for each item, where n is the number of items, and c is the capacity of the knapsack. Amortized running times report the time used for each item, when calculating tolerance limits of all items. Exact tolerance limits are the widest possible intervals, while approximate tolerance limits may be suboptimal. We show how various upper bounds can be used to determine approximate tolerance limits in time $O(\log n)$ or $O(1)$ per item using the Dantzig bound and Dembo–Hammer bound, respectively. The running times and quality of the tolerance limits of all exact and approximate algorithms are experimentally compared, showing that all tolerance limits can be found in less than a second. The approximate bounds are of good quality for large-sized instances, while it is worth using the exact approach for smaller instances.

1. Introduction

In many combinatorial optimization problems the data are not given with certainty, and hence a natural question is how large the errors on the coefficients can be without distorting the sought optimal solution. Combinatorial problems, unlike linear programming problems, behave in an unstable manner under small changes in the initial data, making tolerance analysis a challenging but important problem.

In this paper, we distinguish between sensitivity analysis and tolerance analysis. Sensitivity analysis in linear programming studies in which range the coefficients can vary without changing the current basic solution. Since we do not have basic solutions in combinatorial problems, tolerance analysis studies the robustness of an optimal solution vector to perturbations in the problem coefficients. Tolerance analysis is also known as stability analysis in the literature.

Greenberg (1998) gives a quite recent bibliography for post-optimal analysis in combinatorial optimization, and mentions a number of papers on knapsack problems (Burkard & Pferschy, 1995; Hansen & Ryan, 1996; Kozeratskaya, Lebedeva, & Sergienko, 1983; Seelander, 1980). Klein and Holm (1979) presented a general cutting-plane framework for post-optimal analysis of combinatorial problems and gave sufficient conditions for preserving the same optimal solution when the right-hand side or an objective coefficient is altered.

The *0–1 knapsack problem* consists of packing a subset of n items, each item i having a profit p_i and a weight w_i, into a knapsack of capacity c such that the overall profit is maximized. See, e.g., Kellerer, Pferschy, and Pisinger (2004) for a thorough introduction. Tolerance analysis for the knapsack problem consists of determining the intervals $\alpha_{p_k} \le p_k \le \beta_{p_k}$ and $\alpha_{w_k} \le w_k \le \beta_{w_k}$ for which the profit or the weight of a given item k can be perturbed such that a given optimal solution remains optimal for the problem. Exact tolerance limits are the widest possible intervals, while approximate tolerance limits may be suboptimal (i.e., a subset of the exact tolerance limits). Notice that at any time we only alter a single item k.

Hifi, Mhalla, and Sadfi (2005) proved several results that characterize the tolerance limits. Using these results they proposed two algorithms, one to compute the profit tolerances and one to compute the weight tolerances. The profit algorithm, having a run-

* Corresponding author.
E-mail address: pisinger@man.dtu.dk (D. Pisinger).

http://dx.doi.org/10.1016/j.ejor.2016.10.054
0377-2217/

Table 1
Summary of the results presented by Hifi et al. (2005) and the present paper. Notice that the quality of the approx bounds is different. The approx LP-bound generally gives the most correct tolerance limits of the three approx methods.

Perturbation	Current solution x^*	Limit	Hifi et al. (2005) Exact	Approx	Our results Exact worstcase	Exact amortized	Approx LP-bound	Approx DH-bound
Profit p_k	$x_k^*=0$	α_{p_k}	$O(1)$		$O(1)$	$O(1)$		
		β_{p_k}		$O(n)$	$O(nc)$	$O(c\log n)$	$O(\log n)$	$O(1)$
	$x_k^*=1$	α_{p_k}		$O(n)$	$O(nc)$	$O(c\log n)$	$O(\log n)$	$O(1)$
		β_{p_k}	$O(1)$		$O(1)$	$O(1)$		
Weight w_k	$x_k^*=0$	α_{w_k}		$O(n^2c)$	$O(nc)$	$O(c\log n)$	$O(\log n)$	$O(1)$
		β_{w_k}	$O(1)$		$O(1)$	$O(1)$		
	$x_k^*=1$	α_{w_k}		$O(n^2c)$	$O(nc)$	$O(c\log n)$	$O(\log n)$	$O(1)$
		β_{w_k}	$O(n)$		$O(n)$	$O(1)$		

ning time of $O(n^2)$, applies upper bounds to derive *exact and approximate* tolerance intervals. The weight algorithm, having a running time of $O(n^2c)$, applies dynamic programming to derive *exact and approximate* tolerance intervals.

The main objective of this paper is to present an *exact* algorithm for the tolerance analysis of the 0–1 knapsack problem based on dynamic programming. The algorithm can determine the *exact* tolerance interval for the profit or weight of an arbitrary item.

This approach resembles the approach of Hifi et al. (2005) in the way that both approaches take advantage of the dynamic programming solution, but differs in the fact that some of the results of Hifi et al. (2005) are *approximate* while this new method is *exact* for all results. In addition the new algorithm has a better computation time, $O(nc\log n)$.

Table 1 summarizes the results of Hifi et al. (2005) and of the present paper, reporting the time needed to compute a tolerance limit for a specific item k. The first two rows concern the perturbation of profit p_k while the next two rows concern the perturbation of weight w_k. Columns 4, 6 and 7 report running times for finding *Exact* tolerance limits, while columns 5, 8 and 9 report running times for finding *Approximate* tolerance limits. Depending on the value of the current optimal solution x_k^* the upper and lower limits can be calculated in a variety of ways. All running times are for a given item k, and it is assumed that the current optimal solution is known in advance, including the residual capacity of the solution. *Worstcase* denotes worst-case running time, while *Amortized* denotes amortized running time. Amortized running times report the time used for each item, when calculating tolerance limits of all items. Two different approximate tolerance limits are presented in this paper using either the Dantzig upper bound (*Approx LP-bound*) or Dembo and Hammer (1980) upper bound (*Approx DH-bound*).

Several related problems have been studied recently in the literature: Belgacem and Hifi (2008) and Hifi and Mhalla (2010) consider the perturbation of a subset of items in a binary knapsack problem. Monaci, Pferschy, and Serafini (2013) consider the related robust knapsack problem. Archetti, Bertazzi, and Speranza (2010) consider the reoptimization of a knapsack problem when new items are added to the problem. Various heuristics and approximation algorithms are presented. Monaci and Pferschy (2013) consider a variant of the knapsack problem where the exact weight of each item is not known in advance but belongs to a given interval. The worsening of the optimal solution is analyzed. Plateau and Plateau (2012) consider how a knapsack problem can be reoptimized given that the data has been slightly modified.

The paper is organized as follows: Section 2 describes the 0–1 knapsack problem and its "dual" denoted the *weight knapsack problem*, which is advantageous when determining weight tolerance limits. Dynamic programming methods and upper/lower bounds are presented for both problems. Section 3 formally defines the tolerance analysis of a 0–1 knapsack problem and presents some special cases for which the profit or weight tolerance limits can be identified. Section 4 presents the exact profit and weight tolerance limits, and describes an $O(nc)$ algorithm per item (or $O(n^2c)$ in total) which can be used to calculate the limits. Section 5 shows how the amortized time complexity of the algorithm can be improved to $O(c\log n)$ per item (or $O(nc\log n)$ in total) by making use of overlapping subproblems in the dynamic programming. Moreover, we show how to calculate the tolerance limits by solving a single 0–1 knapsack problem. This makes it possible to use any state-of-the-art algorithm for solving the knapsack problem, and introduces the opportunity to find approximate tolerance limits by use of various upper bounds for the 0–1 knapsack problem.

2. The 0–1 knapsack problem

The 0–1 knapsack problem consists of packing a subset of n items into a knapsack of capacity c. Each item i has profit p_i and weight w_i and the objective is to maximize the profit of the items in the knapsack without exceeding the capacity c. Using the binary variable x_i to indicate whether item i is included in the knapsack, we get the formulation:

$$\begin{aligned}(\text{KP})\ \text{maximize}\ & \sum_{i=1}^{n} p_i x_i \\ \text{subject to}\ & \sum_{i=1}^{n} w_i x_i \le c \\ & x_i \in \{0,1\}, i = 1,2,\dots,n \end{aligned} \quad (1)$$

Without loss of generality we assume that the profits and the weights are *positive* integers (see Kellerer et al., 2004 for transformations to this form). Also, we assume that $\sum_{i=1}^{n} w_i > c$. An optimal solution vector to KP is denoted x^* and the optimal solution value z^*. A knapsack problem with capacity c is denoted KP[c], and we use the terminology KP := KP[c] whenever the capacity is the original capacity. KP[c]\{k} denotes the knapsack subproblem KP[c] where item k is excluded. $z(K)$ is the optimal objective function of knapsack instance K. KP(x') is the instance with variables x fixed at x', hence $z(\text{KP}(x')) = \sum_{i=1}^{n} p_i x'_i$.

The *LP-relaxed* (or fractional) knapsack problem, where $0 \le x_i \le 1$ for $i = 1, 2, \dots, n$ can be solved to optimality by a greedy algorithm, in which the items are sorted according to nonincreasing profit-to-weight ratio p_i/w_i and the knapsack is packed with items $1, 2, \dots$ until the first item s (the *split item*) which does not fit into the knapsack. The optimal solution value z^*_{LP} is then

$$z^*_{LP} = \sum_{i=1}^{s-1} p_i + \left(c - \sum_{i=1}^{s-1} w_i\right)\frac{p_s}{w_s}. \quad (2)$$

Knowing that all profits are integers, we may round down the solution value to $\lfloor z^*_{LP} \rfloor$ getting the *Dantzig upper bound*.