On E-kKP as a knapsack problem related to the conventional 2-approximation algorithm for the 0-1 knapsack problem

Hiroshi Iida*

Abstract

This piece picks up E-kKP as a knapsack problem in relation to the conventional and the simplest 2-approximation algorithm for the 0-1 knapsack problem. Taking account of the similarity between E-kKP and the multiple-choice knapsack problem, we mention how to produce two candidates onto the conventional and also how to obtain an optimal solution of LP-relaxed E-kKP that we require so as to produce the two candidates.

keywords: combinatorial optimization, knapsack problem, approximation algorithm

1 Introduction

We treat E-kKP as a knapsack problem that relates to the conventional 2-approximation algorithm for the classical 0-1 knapsack problem (0-1KP). Before entering the main, we briefly describe 0-1KP and the conventional 2-approximation algorithm for the 0-1KP.

With $N := \{1, 2, \ldots, n\}$, 0-1KP is written as $z^* := \max\{\sum_{j \in N} p_j x_j \mid \sum_{j \in N} w_j x_j \leq c, x_j \in \{0, 1\}\}$ where variable x_j indicates the choice of item $j \in N$

* *E-mail address*: oggi@res.otaru-uc.ac.jp

of two attributes—that is, profit p_j and weight w_j (both are positive integers)—as $x_j=1$ (packed into a knapsack of capacity c)/$x_j=0$ (otherwise). While we call an n-vector of 0-1 variables $x := (x_j)_{j \in N}$ a solution according to the literature, we identify solution x with $S \subseteq N$ as $x_j = 1 \Leftrightarrow j \in S$. Further we call the z^* optimal value, and a solution that gives z^* an optimal solution.

On the other hand, a conventional 2-approximation algorithm for 0-1KP (for the sake of brevity we hereafter call it the conventional) is as follows: after sorting all items in nonascending order of efficiency p_j/w_j, let $s := \min\{k \mid \sum_{j=1}^{k} w_j > c\}$ and we choose the best between $\{1, 2, \ldots, s-1\}$ and $\{s\}$ (i.e., one that has non-smaller value between $\sum_{j=1}^{s-1} p_j$ and p_s). The solution obtained (we hereafter call it 2-approximation solution) fulfills all constraints and has value $\lceil z^*/2 \rceil$ or more. Also, its time complexity is actually the linear time of n, that is, $O(n)$. As regards the performance ratio (guarantee) 2 (i.e., value given by a solution returned is the half of optimal value or more) of the conventional, for the following instance of 0-1KP

j	1	2	3
p_j	2	2	3
w_j	3	3	5
c		5	

$\max\{\sum_{j=1}^{s-1} p_j, p_s\} = 2 \geq \lceil z^*(=3)/2 \rceil$ holds as an equality. In actual fact, the performance ratio 2 is tight. Indeed if we consider the following 0-1KP with huge M as in Kellerer *et al.* [4, p. 34],

j	1	2	3
p_j	2	M	M
w_j	1	M	M
c		$2M$	

then, because value given by a solution returned from the conventional is $M + 2$ and optimal value is $2M$, we have $(M + 2)/2M \to 1/2(M \to \infty)$, which validates the '2' of the 2-approximation is tight and precise as a performance indicator of the conventional.

In the remainder we mention how to produce two candidates that appear when we apply the conventional to E-kKP in Section 2 and also how to obtain an optimal solution of LP-relaxed E-kKP that we need in order to produce the two candidates in Section 3.

2 Two candidates for E-kKP

As a special case of the multi-constrained (multidimensional) knapsack problem [4, Chap 9], we have kKP that has the 2nd constraint $\sum_{j \in N} x_j \le k$ on 0-1KP, which is dealt with in [3] as a knapsack problem related to the conventional. The replacement of the inequality by an equality leads to E-kKP. It is known that we easily have a 2-approximation algorithm for E-kKP by tweaking the one for kKP proposed by Caprara *et al.* [1]—as introduced in [3], it's similar to the conventional—named $H^{1/2}$ (Kellerer *et al.* call it LP-Approx [4, Fig 9.3]) [4, Subsect 9.7.4].

Concretely, since in an LP-solution (an optimal solution of linear programming relaxed problem, i.e. admitting $0 \le x_j \le 1$ as a relaxation of $x_j \in \{0, 1\}$) of E-kKP there are 0 or 2 variables of $x_j \notin \{0, 1\}$ fractional (in kKP having the same number of constraints as E-kKP, it's 2 or less; however, $\sum_j x_j = k$ eliminates the case of 1. Then, in the case where there is no fractional variable in an LP-solution obtained, the algorithm returns the LP-solution optimal; otherwise, supposing two fractional variables, say i, j $(w_i < w_j)$[1] the

[1] If $w_i = w_j$, without considering the combination of $x_i + x_j = 1$, we can augment

best of the following two gives a 2-approximation solution.

1. A set being made up of all items of = 1 in the LP-solution, the number of which is $k-1$, and item i lighter: This candidate is the same as that of the multiple-choice knapsack problem (MCK [4, Chap 11]). In fact, the number of fractional variables in an LP-solution of MCK is also 0 or 2; nonetheless, the next one is different from that of MCK.
2. A set comprising the lightest $k-1$ items among $N\setminus\{j\}$ and item j heavier

Like this, due to $\sum_j x_j = k$, the 2nd candidate including the heavier j is constructed in a different way of $H^{1/2}$—Differing from kKP that admits a solution including item j only, E-kKP is a little bit similar to MCK in which any solution has a fixed cardinality, viz. equal to the number of classes (we must select only one item in each class). More precisely in MCK, on the 2nd candidate, items added to the heavier j are the lightest item in each class except a class extracting the j [4, p. 338].

value given by the LP-solution with taking an item of more valuable; thus, it implies $p_i = p_j$ too. Therefore without considering the combination of the two same items, we can set all variables of the LP-solution to 0-1 by taking either item only. In consequence setting $w_i < w_j$ doesn't loose the generality. This will also be the case as for E-kSSP (an E-kKP of $p_j = w_j$, $\forall_j$) or kKP. We can further set $p_i < p_j$, because $p_i \geq p_j$ and $w_i < w_j$ have made a chance to consider $x_i = 1$, $x_j = 0$ (see Lemma 9.7.2 in Kellerer *et al.* [4, p. 277] — $I = \{i, j\}$ that appeared therein is a tiny misprint and should be $F = \{i, j\}$. Moreover at the beginning of its proof, the description of "By definition of I" should be "By definition of F").

3 How to solve an LP-relaxed E-kKP

As in Section 2, when we apply the conventional to E-kKP given, we need an LP-solution of the E-kKP in the same way as kKP. Can we solve LP-relaxed E-kKP in easier way like MCK? As we have seen, E-kKP is similar to MCK. In particular, (albeit it's obvious) E-kKP of $k=1$ has the same structure as MCK of one class only. Taking an algorithm for LP-relaxed MCK into account, the following one for LP-relaxed E-kKP will naturally be drawn (for more details around how to solve LP-relaxed MCK, see, e.g., Iida [2]).

First of all we sort all items in ascending order of weight, and let $K := \{1, 2, \ldots, k\}$ and $\bar{K} := N \setminus K$. Following, $w(K)$ means $\sum_{j \in K} w_j$. In MCK, an initial set for solving LP-relaxed problem is one including the lightest item in each class, which corresponds to K. After plotting all items onto a plane with x-axis indicating weight and y-axis profit, we consider a bipartite graph consisting of K and $\bar{K}$. As an edge, from each element in K, if there is an item in $\bar{K}$ of more valuable and non-lighter than the element then we connect the two. If there are plural candidates in $\bar{K}$ for connecting with some item in K then we select the largest gradient among those. After the preparation above, we iterate the following operation until $w(K) > c$. Namely, this operation corresponds to the exchange of items along a slope in a class on MCK.

Operation: Choose an edge $(i, j) \in K \times \bar{K}$ of the largest gradient among at most k edges and exchange i and j between K and $\bar{K}$, that is, $K := (K \setminus \{i\}) \cup \{j\}$. According to this, we make edges up-to-date concerning new K, $\bar{K}$. Specifically we connect a new $j \in K$ with an item in $\bar{K}$ if possible. In addition, on an item in K connected to $j \in \bar{K}$

removed, we provide a new edge from the item to $\bar{K}$ if possible. Moreover if there is an item in K such that we can provide a new edge or can augment a gradient by connecting with new $i \in \bar{K}$, we connect or update the one in K with the new i.

If we have no edge against $w(K) \leq c$, we have the most valuable set of cardinality k within c; thus, it's optimal (we will mention it afterward); otherwise we stop by $w(K) > c$, suppose an edge corresponding to the last exchange is (i, j). On K just before exceeding c, including i, $x_q = 1$ for all $q \in K \setminus \{i\}$ and

$$\begin{cases} x_i = (w(K) - w_i + w_j - c)/(w_j - w_i) \\ x_j = 1 - x_i = (c - w(K))/(w_j - w_i) \end{cases}$$

(otherwise = 0) will be an LP-solution of E-kKP. Do we lose something?

For instance we consider $k = 2$, $c = 5$ in Fig 1. In this example, K moves $\{1, 2\} \rightarrow \{1, 3\} \rightarrow \{2, 3\}$ and reaches the optimal (the most valuable k items), consuming all edges at last. One more instance, we consider $k = 2$, $c = 9$ in Fig 2. Starting at $K = \{1, 2\}$, we move $\rightarrow \{1, 3\} \rightarrow \{1, 4\} \rightarrow \{3, 4\}$ and gain

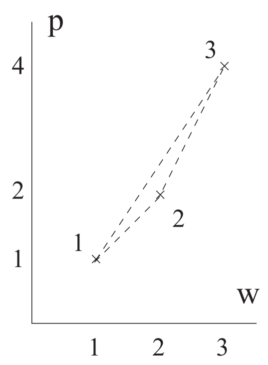

Figure 1: Example 1 of E-kKP ($n = 3$)

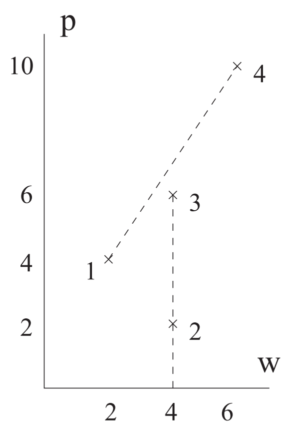

Figure 2: Example 2 of E-kKP ($n = 4$)

value 15 given by $x_1 = x_3 = 1/2$, $x_4 = 1$ (a solution of $x_1 = 1/4$, $x_3 = 1$, $x_4 = 3/4$ gives 14.5 below it).

In what follows we define a gradient of an edge $(i, j) \in K \times \bar{K}$ as an angle between the vector (i, j) and the x-axis representing weight. As in Example 2, a gradient just $\pi/2$ appears in the case where the kth item has the same weight as the $(k+1)$-st item when we select the k lightest items as an initial K. Then, does an edge of a gradient greater than or equal to $\pi/2$ appear during operations? Why do we think about such a thing? The reason for which is that we assume $w(K)$ increases by an exchange of items. Under the assumption, we can exit at $w(K) > c$ in short order. In MCK, for example, there are no two items of the same weight in a class (an item of more valuable remains in the two of the same weight), and the weight strictly increases by an exchange of items along a slope.

Then during operations, we assume that such an edge (i, j) in Fig 3 appears for the first time. If j has been in $\bar{K}$ from the initial stage, it implies that item i heavier than j was into K as a result of some exchange. However, considering an edge corresponding to the exchange, an item on the K side (by the assumption, it's lighter than j) produces a larger gradient by connecting not i but j. Therefore an exchange should firstly be done with not i but j and j has been into K before i entering K. Thus at the stage of $i \in K$, it's hard to claim that j has been in $\bar{K}$ from the initial stage and no exchange as to j has been done. As a consequence we can contend that for $i \in \bar{K}$, an exchange that removes j from K shall be done. Here let an edge corresponding to the exchange be (j, q) (by the assumption, q is heavier than i). According to the same argument as the previous, since the gradient of (i, q) is greater than that of (j, q), an exchange as to (i, q) should firstly be done, and an exchange as to (j, q) must not be done under $i \in K$. Consequently, it will be possible to conclude that the edge (i, j) cannot

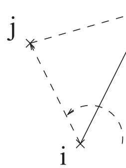

Figure 3: Does an edge of a gradient > $\pi/2$ appear?

appear.

Here, in another view, we take a look at the argument hitherto with regard to the gradient $\pi/2$ or more again. Before any exchange, the elements of K and $\bar{K}$ are divided into the left hand side and the right hand side, respectively. As in Fig 3, $i \in K$ and $j \in \bar{K}$ imply that an exchange as to i or j has been done. If, in an initial stage, $j \in K$ and $i \in \bar{K}$; then, it's impossible to connect j with i of less valuable than j; thus, either stage of $i, j \in K$ or $i, j \in \bar{K}$ should be passed through before reaching Fig 3. Therefore in the same argument as the previous: in the case of $i, j \in \bar{K}$, before i is in K, j shall be in K; in the case of $i, j \in K$, before j is removed from K, i shall be removed from K, I guess.

References

[1] Alberto Caprara, Hans Kellerer, Ulrich Pferschy and David Pisinger, Approximation algorithms for knapsack problems with cardinality constraints. *European J Oper Res* **123**, 333-45 (2000) http://dx.doi.org/10.1016/S0377-2217(99)00261-1.

[2] Hiroshi Iida, Towards an alternative relaxation for the multiple-choice knapsack problem. *Economic Review* **52**.4, 267-82 (Otaru Univ of Commerce, 2002) http://hdl.handle.net/10252/496.

[3] Hiroshi Iida, A further addendum to "Some thoughts on the 2-approximation algorithm for knapsack problems: a survey." Discussion paper series no. 167, CBC at Otaru Univ of Commerce, 2014; http://hdl.handle.net/10252/5386.

[4] Hans Kellerer, Ulrich Pferschy and David Pisinger, *Knapsack Problems*. Springer 2004.