

Regarding the Failure of Applying the Conventional 2-approximation Algorithm to the Collapsing Knapsack Problem

Iida, Hiroshi*

¹ Otaru Univ. of Commerce, Japan.

*Corresponding author: oggi@res.otaru-uc.ac.jp

Abstract:

We show that the conventional 2-approximation algorithm for the classical 0–1 knapsack problem does not work for the collapsing knapsack problem in general. We also show that the algorithm will work for the problem under some special conditions.

Keywords:

Combinatorial Optimization; Collapsing Knapsack Problem; 2-approximation Algorithm

In this report we deal with the collapsing knapsack problem (hereafter CKP [1, pp. 416–9]), which is an extension of the classical 0–1 knapsack problem (KP), and the conventional 2-approximation algorithm for the KP.

The KP is one of well-known and \mathcal{NP} -hard combinatorial optimization problems. In the KP, given n items of profit and weight, we pack the items into a knapsack of capacity c so that the total profit of packed items is maximized without the total weight of those exceeding the capacity c . With $N := \{1, 2, \dots, n\}$ we can formulate KP as $z_{\text{KP}} := \max\{\sum_{j \in N} p_j x_j \mid \sum_{j \in N} w_j x_j \leq c, x_j \in \{0, 1\}\}$ where the 0–1 variable x_j indicates the choice of item $j \in N$ as $x_j = 1$ (packed)/ $x_j = 0$ (otherwise). In what follows, for all item j we assume that both profit p_j and weight w_j are positive integers and $w_j \leq c$. In addition we call a subset of N a solution and for solution $S (\subseteq N)$ we call $\sum_{j \in S} p_j$ (resp. $\sum_{j \in S} w_j$) the profit (resp. the weight) of S . Moreover a solution of weight within c is said to be feasible.

A 2-approximation algorithm for KP will be able to be defined as one that returns a feasible solution whose profit is the half of optimal value z_{KP} or more. By this we can state the conventional 2-approximation algorithm for KP as follows: after sorting all items in nonascending order of efficiency (profit-to-weight ratio)—that is, $p_1/w_1 \geq p_2/w_2 \geq \dots \geq p_n/w_n$ —let $k := \min\{\ell \mid \sum_{j=1}^{\ell} w_j > c\}$. Then, the algorithm returns a more profitable solution between $\{1, 2, \dots, k-1\}$ and $\{k\}$; that is, $\max\{\sum_{j=1}^{k-1} p_j, p_k\} > z_{\text{KP}}/2$. A point is that we require two feasible solutions of which the profit sum is z_{KP} or more. For further details, see, for example, Korte and Vygen [2, p. 462].

The CKP, whose knapsack will collapse according to the number of packed items, is an extension of KP. For example we have a fragile item and each shall be covered with something when packed, so the larger the number of packed items, the smaller the capacity of the knapsack, due to the covering of each item. The capacity of KP is a constant while on CKP, it is a functional over N such that $c(\sum_{j \in N} x_j)$ where

$c(1) \geq c(2) \geq \dots \geq c(n)$. In the following we assume $w_j \leq c(1)$ for all $j \in N$. In addition, for CKP, the k included in the conventional 2-approximation algorithm will be $\min\{\ell \mid \sum_{j=1}^{\ell} w_j > c(\ell)\}$. $w_j > 0$ and the monotonicity of $c(\cdot)$ lead to $\sum_{j=1}^{k+1} w_j > \sum_{j=1}^k w_j > c(k) \geq c(k+1)$; thus, the k for CKP is well-defined. We hereafter call the conventional 2-approximation algorithm for KP, the conventional, for short.

Unfortunately, the conventional does not work for CKP. The following instance of CKP validates our claim.

j	1	2	3
p_j	3	4	9
w_j	2	3	8
$c(j)$	8	4	

There nonetheless exist a few cases of the conventional working for CKP. In the rest of this paragraph we assume $p_1/w_1 \geq p_2/w_2 \geq p_j/w_j$ for all $j \geq 3$. In fact, when $w_1 + w_2 > c(1) (\geq c(2))$ the greater between p_1 and p_2 is beyond the half of optimal value. Moreover, when $w_1 + w_2 = c(1) = c(2)$ it also works, because $p_1 + p_2$ is optimal value. Needless to say $c(1) = c(2) = \dots = c(n)$ or $w_1 = c(1)$ is the case, too.

Here we would like to add that a KP with a condition

$$p_j = w_j \text{ for any } j \in N \tag{1}$$

is called the subset-sum problem (SSP). The SSP is a well-studied problem too, and has a special structure such that the efficiency of items is constant ($\equiv 1$). By this, without sorting items, the conventional works for SSP. Indeed, when $\sum_{j=1}^k w_j > c$ with minimum k , the heavier of two feasible solutions—that is, $\{1, 2, \dots, k-1\}$ and $\{k\}$ —is heavier than $c/2 \geq z_{\text{SSP}}/2$ where $z_{\text{SSP}} := \max\{\sum_{j \in N} w_j x_j \mid \sum_{j \in N} w_j x_j \leq c, x_j \in \{0, 1\}\}$.

Then, what does the condition (1) give to CKP? In the remaining we investigate a CKP of (1), which we shall call collapsing subset-sum problem (CSSP). In applying the conventional to CSSP, as same as the case of SSP, we need not sort items according to the efficiency of items.

If there exists no restriction, the following instance of CSSP claims that the conventional does not work for CSSP, where we have $k = 2$ and $\{1\}$ has weight 4 as same as that of $\{2\}$; however, optimal value is $9 > 4 \times 2$.

j	1	2	3
w_j	4	4	9
$c(j)$	9	7	

Then, how about applying the conventional to a CSSP in which all weights are sorted in nonascending order? If $\sum_{j=1}^k w_j (> c(k))$ is below optimal value, the optimal value is the weight sum of more than k items; but, the weight sum of more than k items is at most $c(k+1)$ and $\sum_{j=1}^k w_j > c(k) \geq c(k+1)$. Therefore $\sum_{j=1}^k w_j$ is heavier than any solution whose cardinality is greater than k , so it is a contradiction.

Finally we would like to add a concluding remark: we claim that the conventional will work for CSSP, provided that weights are sorted in nonascending order beforehand.

Acknowledgments

Thanks are due to two anonymous reviewers for their valuable comments, which definitely improved the presentation of this report.

References

- [1] H. Kellerer, U. Pferschy and D. Pisinger, *Knapsack Problems*. Springer Berlin 2004.
- [2] B. Korte and J. Vygen, *Combinatorial Optimization*. 5th Edition, Springer Berlin 2012.