A survey of reductions among knapsack problems

Hiroshi Iida
e-mail: *oggi@res.otaru-uc.ac.jp*

abstract

This paper deals with several reductions among knapsack problems. To devise a reduction from another problem to a given problem is a usual technique to determine the computational complexity of the given problem while we discuss reductions which are employed in order to solve given problems. Reductions on three types of variants of the 0-1 knapsack problem are included.

Keywords: knapsack problem; strongly correlated knapsack problem; subset-sum problem; partition problem; collapsing knapsack problem

1 Introduction

A reduction from another problem to a given problem is frequently employed to describe the computational complexity of the given problem as in Karp [11]. On knapsack problems, e.g. the maximum clique problem is reduced to the set-union knapsack problem in Goldschmidt et al [6], and the set covering problem is reduced to the max-min 0-1 knapsack problem in Yu [20]. On the other hand, a reduction is also employed to solve a given problem in a framework already studied, that is, producing another known problem by the reduction, and solving it by an algorithm already developed. This paper gathers such reductions on three types of variants of the 0-1 knapsack problem.

The 0-1 Knapsack Problem (KP) is a typical combinatorial optimization problem. In the KP, where items and a knapsack are given, we pack the items into the knapsack so that the total profit of the packed items is maximized without exceeding the capacity of the knapsack. The KP is stated as follows:

$$\text{(KP)} \quad \text{maximize} \quad \sum_{j=1}^{n} p_j x_j$$

$$\text{subject to} \quad \sum_{j=1}^{n} w_j x_j \leq c$$

$$x_j \in \{0, 1\}, \ j = 1, 2, \ldots, n,$$

where the profit p_j and weight w_j for any item j and the capacity c are all positive integers. We will note that the term "profit-to-weight ratio" frequently used in this paper indicates p_j / w_j. Without loss of generality we will assume that $w_j \leq c$ for all j and $\Sigma_{j=1}^{n} w_j > c$ in order to exclude unpromising items and trivial problems respectively. Throughout this paper, the set $\{1, 2, \ldots, n\}$ is denoted by N. In addition we sometimes identify a set $J \subset N$ with a solution vector $(x_j)_{j \in N}$ as $j \in J \Leftrightarrow x_j = 1$. For details on KP, see, e.g. Martello and Toth [12], and Chapter 13 in Ibaraki and Fukushima [7] in Japanese.

The remainder of this paper is organized as follows: Each of Sections 2-4 is a case study. In Section 2, the strongly correlated knapsack problem is discussed. The subset-sum problem is in Section 3, and the collapsing knapsack problem is in Section 4. The final section is devoted to conclusion.

2 Strongly correlated knapsack problem

The Strongly Correlated 0-1 Knapsack Problem (SCKP) is a special case of KP. In the SCKP, the profit p_j of each item j is given by $w_j + k$ where the fixed-charge k is a positive constant. The SCKP is stated as follows:

$$\text{(SCKP)} \quad \text{maximize} \quad \sum_{j \in N} (w_j + k) x_j \quad (k > 0)$$

$$\text{subject to} \quad \sum_{j \in N} w_j x_j \leq c \tag{1}$$

$$x_j \in \{0, 1\}, \ j \in N.$$

While an algorithm for KP can be applied to SCKP, the SCKP is hard to solve due to having a rather narrow span of profit-to-weight ratios as stated in Balas and Zemel [2].

Example 1. Consider an instance of SCKP with eight items. The weights are given as follows:

j	1	2	3	4	5	6	7	8
w_j	2	4	8	12	20	28	46	72,

and $k = 10$, $c = 100$. It should be noted that, under the assumption $w_j \leq w_{j+1}$, it follows that $(w_j + k)/w_j \geq (w_{j+1} + k)/w_{j+1}$.

We will here attempt to fix $x_1 = 1$. First, as an initial solution obtained by means of an ordinary greedy heuristic we have {1, 2, 3, 4, 5, 6} of profit-sum 134, since $\Sigma_{j=1}^{6} w_j = 74 \leq c < 120 = \Sigma_{j=1}^{7} w_j$. Next, by the linear programming relaxation problem of the given instance of SCKP with $x_1 = 0$ we obtain the Dantzig upper bound (Dantzig [4])

$$\left\lfloor 14 + 18 + 22 + 30 + 38 + (100 - 72) \times \frac{56}{46} \right\rfloor = 156 > 134.$$

Consequently we cannot fix x_1 despite a maximum profit-to-weight ratio.

Recently, a specialized algorithm incorporating a reduction of SCKP was proposed by Pandit and Ravi Kumar [14]. The point of the reduction is that the objective function of (1) can be viewed as

$$\sum_{j \in N} w_j x_j + \beta k,$$

where the β indicates the number of packed items. By this, the problem (1) is equivalent to the group of subset-sum problems each of which is of an additional constraint on the cardinality of an optimal solution. As will be discussed in the next section, the subset-sum problem is a special case of KP, where $p_j = w_j$ for all $j \in N$. The problem equivalent to (1) is stated as follows:

$$\begin{aligned} \text{maximize} \quad & \sum_{j \in N} w_j x_j + \beta k \\ \text{subject to} \quad & \sum_{j \in N} w_j x_j \leq c \\ & \sum_{j \in N} x_j = \beta \leq \text{UB} \\ & x_j \in \{0, 1\}, \ j \in N, \end{aligned} \tag{2}$$

where $\text{UB} := \max\{\, i \mid \Sigma_{j=n-i+1}^{n} w_j \leq c \,\}$ under the assumption that $w_1 \geq w_2 \geq \cdots \geq w_n$. The UB indicates an upper bound of the number of packed items, that is, if we pack more than UB items into the knapsack then it always turns out an infeasible solution. The point of SCKP is that the larger the number of packed items, the larger the number of the fixed-charge k also contained in the knapsack. Therefore, the process to find an optimal solution of (2) should be in descending order of the cardinality of a solution, e.g. in the Pascal programming language: **for** $\beta :=$ UB **downto** 1.

The above equivalency is also used in succeeding studies on SCKP: Pisinger [17] and Iida [9]. On the other hand, Martello and Toth [13] proposed an algorithm for KP but also applicable to SCKP. The algorithm proposed in [17] named SCKNAP solves equivalent subset-sum problems by dynamic programming while the one in [9] named XSC employs branch-and-bound approach. Also, the SCKNAP involves a promising heuristic which attempts to find an optimal solution under the condition $\beta = \text{UB}$. A comparison of SCKNAP and the algorithm in [13] is presented in [17], and the two algorithms SCKNAP and XSC were compared in [9]. Summarizing in a word, the SCKNAP shows the best performance among the three.

3 Subset-sum problem

In this section we discuss a reduction of the Subset-Sum Problem (SSP), focusing on an approach by Iida and Vlach [8]. To begin with we should like to state that, as we have seen in the previous section, the SSP is used to solve SCKP. In addition, the SSP is also used in Pisinger [18] to solve the multiple knapsack problem. The necessity to solve the SSP thus arises in several situations. Hence, not only in a theoretical but also a practical point of view, it is never meaningless to study an algorithm for SSP. Here we will formally state the SSP.

$$
\begin{aligned}
\text{(SSP)} \qquad & \text{maximize} \quad \sum_{j \in N} a_j x_j \\
& \text{subject to} \quad \sum_{j \in N} a_j x_j \leq c \\
& \qquad\qquad\quad x_j \in \{0, 1\}, \; j \in N.
\end{aligned}
$$

Since the SSP is a special case of KP as mentioned in the previous section, an algorithm for KP can be applied to SSP. The branch-and-bound algorithm for KP however shows catastrophic behavior when applied to SSP, so it is similar to complete enumeration. This fact was reported by Ahrens and Finke [1]. On the SSP, the point is that the Dantzig upper bound is always equal to the capacity c due to the constant profit-to-weight ratio equal to one. For further details on SSP, see, e.g. Chapter 4 in [12], and Chapter 10 in Pisinger [16]: A tailored algorithm is also proposed in each.

In [8], a given instance of SSP is reduced to the Partition Problem (PP). The PP is stated as follows: Given n positive integers $c_1, \ldots, c_n$; find J a subset of N so that it minimizes $| \Sigma_{j \in J} c_j - \Sigma_{j \in N \setminus J} c_j |$. By the definition, it is clear that if J is optimal then $N \setminus J$ is also optimal.

The reduction from SSP with n items $a_1, \ldots, a_n$ and capacity c to PP with $n+2$ items $c_1, \ldots, c_{n+2}$ is defined by

$$\begin{cases} c_j &= a_j, \qquad \text{for all } j \in N \\ c_{n+1} &= 2c \\ c_{n+2} &= \Sigma_{j \in N} a_j. \end{cases} \tag{3}$$

Note that, for an optimal solution of the PP (3), it contains either item $n+1$ or $n+2$: Among all solutions containing both items $n+1$ and $n+2$, a minimized objective value is obviously $2c$ achieved by $\{n+1, n+2\}$. Also, the objective value of the set $\{1, n+2\}$ is $2(c-a_1)$, since $c \geq a_1$. Then, the positiveness of a_1 completes the argument, that is, we have $2(c-a_1) < 2c$.

The reduction (3) suggests that, for some $J \subset N$, if the set $J \cup \{n+2\}$ is optimal to the PP (3) then the set J will solve the original SSP. Indeed, if the sum of c_j corresponding to the set $J \cup \{n+2\}$ is exactly the half of the overall sum of c_j's in (3), i.e. $c + \Sigma_{j \in N} a_j$, then $\Sigma_{j \in J} a_j = c$ follows.

Example 2. Consider an instance of SSP given by $n=3$, $a_1=3$, $a_2=4$, $a_3=8$ and $c=9$. Using the reduction (3) we obtain the following instance of PP: $c_1=3$, $c_2=4$, $c_3=8$, $c_4=18$, $c_5=15$. The optimal solution containing item $n+2$ is $\{3, 5\}$, and $a_3=8 \leq 9=c$. Then, $\{3\}$ is feasible for the original instance of SSP, and indeed optimal.

Example 3. Consider an instance of SSP given by $n=3$, $a_1=3$, $a_2=4$, $a_3=8$ and $c=10$. Similarly we obtain PP: $c_1=3$, $c_2=4$, $c_3=8$, $c_4=20$, $c_5=15$. The optimal solution containing item $n+2$ is $\{1, 3, 5\}$, however, $a_1+a_3=11>10=c$. Therefore, $\{1, 3\}$ is infeasible for the original instance of SSP, and it cannot be optimal.

As implied by Example 3, the optimal solution of the PP (3) does not always deliver an optimal solution of the original SSP. The failure arises in the case where

$$\sum_{j \in J \cup \{n+2\}} c_j > \sum_{j \in (N \setminus J) \cup \{n+1\}} c_j, \qquad J \subset N.$$

In light of this, the reduction (3) is not completed. As a result, an algorithm employed in [8] in order to solve the PP (3) was slightly modified to ensure the optimality of a finally yielded solution.

A comparison of the algorithms proposed in [12], [16], and [8] respectively is presented in [8]. In conclusion, the algorithm in [8] has outperformed the others only for a few hard instances: problems Todd and Avis, the definitions of which are in Chvátal [3].

4 Collapsing knapsack problem

Posner and Guignard [19] introduced a more complicated problem than KP named Collapsing 0-1 Knapsack Problem (CKP) with a nonconstant capacity. In the CKP, the knapsack will collapse according to the number of packed items. For instance, each item is an antique, and should be covered with something strong respectively when packed. Then the larger the number of packed items, the smaller the capacity of the knapsack, due to the strong covering each item. The CKP is stated as follows:

$$\begin{aligned} \text{(CKP)} \quad & \text{maximize} \quad \sum_{j \in N} p_j x_j \\ & \text{subject to} \quad \sum_{j \in N} w_j x_j \le b\Big(\sum_{j \in N} x_j \Big) \qquad (4) \\ & \qquad x_j \in \{0,1\}, \ j \in N, \end{aligned}$$

where the $b(\cdot)$ is a given monotone nonincreasing function on the discrete domain N as $b(1) \ge b(2) \ge \cdots \ge b(n)$.

In the literature several algorithms for CKP have been proposed, e.g. Fayard and Plateau [5], and Pferschy et al [15]. In particular, Pferschy et al [15] proposed two simple but efficient algorithms, in which we discuss the first one in this section.

The first algorithm incorporates a reduction which produces an instance of KP equivalent to a given CKP, and solves the resulting instance by means of an algorithm for KP. Based on the CKP (4), the reduction constructs KP with $2n$ items of weights $\alpha_1, \ldots, \alpha_{2n}$ and profits $\gamma_1, \ldots, \gamma_{2n}$. The coefficients are defined as follows:

$$\alpha_j = \begin{cases} w_j + A & \text{for } j = 1, \ldots, n \\ (4n-j)A - b(j-n), & \text{for } j = n+1, \ldots, 2n \end{cases}$$
$$\gamma_j = \begin{cases} p_j + C & \text{for } j = 1, \ldots, n \\ (3n+1-j)C, & \text{for } j = n+1, \ldots, 2n, \end{cases} \tag{5}$$

where $A = \Sigma_{j \in N} w_j$ and $C = \Sigma_{j \in N} p_j$. Then the resulting KP, which we will call SKP, is stated as follows:

$$\begin{aligned} \text{(SKP)} \quad & \text{maximize} \quad \sum_{j=1}^{2n} \gamma_j x_j \\ & \text{subject to} \quad \sum_{j=1}^{2n} \alpha_j x_j \leq B \\ & \qquad x_j \in \{0, 1\}, \ j = 1, \ldots, 2n, \end{aligned} \tag{6}$$

where the capacity B is defined as $3nA$. Following the terminology in [15] we will hereafter call an item with index in N *small item* and an item with index in $\{n+1, \ldots, 2n\}$ *large item*. The following validates the equivalency of a given CKP and the SKP (6).

Theorem (Pferschy et al, 1997). The instance of CKP has a feasible solution with objective value V if and only if the instance of SKP has a feasible solution with objective value $V+(2n+1)C$.

Proof. see [15].

It should be noted that on a feasible solution of SKP the contents of the knapsack comprise one large item j and $j-n$ small items. In another view, the choice of a large item determines the number of small items packed.

On the other hand, the second algorithm in [15] is tailored for CKP. In fact, the first algorithm incorporating the above reduction is outperformed by the second, one reason for which is due to the huge coefficients appeared. Further, the coefficients of small items are highly correlated. More precisely, the profit-to-weight ratio of any small item is almost equal to C/A, which also makes the resulting instance hard to solve.

At the end of this section we would like to add that the coefficients defined by (5) can be replaced with the following, provided all items in CKP cannot be packed, viz. $b(n) < \Sigma_{j \in N} w_j$.

$$\alpha_j = \begin{cases} w_j + A & \text{for } j = 1, \dots, n \\ (3n-j)A - b(j-n), & \text{for } j = n+1, \dots, 2n-1 \end{cases}$$

$$\gamma_j = \begin{cases} p_j + C & \text{for } j = 1, \dots, n \\ (2n+1-j)C, & \text{for } j = n+1, \dots, 2n-1, \end{cases}$$

where the capacity B is $2nA$. Moreover, the $V+(2n+1)C$ in the statement of Theorem is replaced with $V+(n+1)C$. The contents of the knapsack on a feasible solution of SKP however remain unchanged regardless of the replacement as one large item j and $j-n$ small items.

It will seem preferable that the replacement contributes to the decrease of the magnitude of the coefficients. In a practical point of view however the revised one will not improve the performance of the algorithm employed for SKP, since there exists no change to the remaining weight for $j-n$ small items after the choice of a large item j. For further details, see Iida [10].

5 Conclusion

In this paper we have discussed three reductions each of which is employed to solve a variant of the 0–1 knapsack problem. In summary: First, as implied by the case that the SCKP is reduced to the group of SSP's, a hard problem could be solved by being reduced to another problem. This approach, which could generally be called a reduction approach, has still been involved in the state-of-the-art algorithm for SCKP. Second, although the reduction approach for SSP merely resulted in showing a rather limited efficiency, a few hard instances of SSP were solved more efficiently than the other algorithms due to the reduction from SSP to PP. Third, a problem could be reduced to a more simple one than the original as the CKP, which is an extension of KP, is indeed reduced to the KP, thus a reduction itself is fairly interesting to study. Furthermore, a reduction approach mostly implies that it is not necessary to develop an algorithm tailored, which also makes it attractive. Finally, we hope that this paper will drop a hint to solve another variant of the 0–1 knapsack problem in quite a novel way.

References

[1] J. H. Ahrens and G. Finke, "Merging and sorting applied to the zero-one knapsack problem," *Operations Res.* **23**(6) 1099–1109 (1975).

[2] E. Balas and E. Zemel, "An algorithm for large zero-one knapsack problems," *Operations Res.* **28**(5) 1130–1154 (1980).

[3] V. Chvátal, "Hard knapsack problems," *Operations Res.* **28**(6) 1402–1411 (1980).

[4] G. B. Dantzig, "Discrete-variable extremum problems," *Operations Res.* **5**(2) 266–277 (1957).

[5] D. Fayard and G. Plateau. "An exact algorithm for the 0-1 collapsing knapsack problem," *Discrete Appl. Math.* **49**, 175–187 (1994).

[6] O. Goldschmidt, D. Nehme and G. Yu, "Note: On the set-union knapsack problem," *Naval Res. Logistics* **41**(6) 833–842 (1994).

[7] T. Ibaraki and M. Fukushima『最適化プログラミング』Iwanami, Tokyo, Japan: First impression, 1991.

[8] H. Iida and M. Vlach, "An exact algorithm for the subset-sum problem," *The Institute of Statistical Mathematics Cooperative Research Report* (統計数理研究所共同研究リポート) *92 Optimization – Modeling and Algorithms* – **9**, 11–29 (1996).

[9] H. Iida, "A simple branch-and-bound approach for the strongly correlated knapsack problem," *Economic Review* (商学討究) **48**(2/3) 353–370, Otaru University of Commerce (1998).

[10] H. Iida, "A short note on the reducibility of the collapsing knapsack problem," submitted to *J. Opns. Res. Soc. Japan*, No. 00-09 (2000) under review.

[11] R. M. Karp, "Reducibility among combinatorial problems," in R.E. Miller and J. W. Thatcher (eds.) *Complexity of Computer Computations*, Plenum Press, New York (1972) 85–103.

[12] S. Martello and P. Toth, *Knapsack Problems: Algorithms and Computer Implementations*, John Wiley & Sons, Chichester, England, 1990.

[13] S. Martello and P. Toth, "Upper bounds and algorithms for hard 0-1 knapsack problems," *Operations Res.* **45**(5) 768–778 (1997).

[14] S. N. N. Pandit and M. Ravi Kumar, "A lexicographic search for strongly correlated 0-1 knapsack problems," *Opsearch* **30**(2) 97–116 (1993).

[15] U. Pferschy, D. Pisinger and G. J. Woeginger, "Simple but efficient approaches for the collapsing knapsack problem," *Discrete Appl. Math.* **77**, 271–280 (1997).

[16] D. Pisinger, *Algorithms for Knapsack Problems*, Ph.D. thesis, Dept. of Computer Science, University of Copenhagen (DIKU), Denmark, 1995.

[17] D. Pisinger, "A fast algorithm for strongly correlated knapsack problems," *Discrete Appl. Math.* **89**, 197–212 (1998).

[18] D. Pisinger, "An exact algorithm for large multiple knapsack problems," *Euro. J. Opnl. Res.* **114**(3) 528–541 (1999).

[19] M. E. Posner and M. Guignard, "The collapsing 0–1 knapsack problem," *Math. Programming* **15**(2) 155–161 (1978).

[20] G. Yu, "On the max-min 0-1 knapsack problem with robust optimization applications," *Operations Res.* **44**(2) 407–415 (1996).