

## DP-BASED ALGORITHM AND FPTAS FOR THE KNAPSACK SHARING AND RELATED PROBLEMS

Seiji Kataoka  
*National Defense Academy*

Takeo Yamada  
*National Defense Academy, Emeritus*

(Received February 6, 2018; Revised September 18, 2018)

*Abstract* In the *knapsack sharing problem* (KSP), formulated previously, we considered a game-theoretic situation in which two or more players (agents) compete for their share of capacity in a knapsack with their respective sets of items. As an extension of this problem, we formulate the *extended knapsack sharing problem* (XKSP). This is actually a family of KSP-like problems, and we present a *dynamic programming*-based (DP-based), pseudo-polynomial time algorithm to solve XKSP to optimality in a unified way. XKSP is shown to be  $\mathcal{NP}$ -hard, but due to the existence of this pseudo-polynomial time algorithm, it is only weakly  $\mathcal{NP}$ -hard. Next, we develop an algorithm to solve the problem approximately in polynomial time by decomposing it into a series of subproblems. Furthermore, we introduce a scaling factor into the DP computation to obtain a *fully polynomial time approximation scheme* (FPTAS) for XKSP with two agents. Extension to the case of more than two agents is discussed, together with a non-DP-based PTAS.

**Keywords:** Combinatorial optimization, knapsack sharing problem, DP-based algorithm, approximation, FPTAS/PTAS

### 1. Introduction

In this article, we address a game-theoretic situation in which  $K$  players (or *agents*) compete for their share of capacity in a knapsack with their respective sets of items.  $N_k = \{1, 2, \dots, n_k\}$  denotes the index set of items belonging to agent  $k$ , and the *weight* and *profit* of item  $j \in N_k$  are denoted  $w_{kj}$  and  $p_{kj}$ , respectively. The aim is to pack the knapsack with these items within the capacity limit  $C$  in a way that is satisfactory for all agents as a whole. Let  $x_{kj} \in \{0, 1\}$  be a decision variable that takes the value 1 if item  $j \in N_k$  is accepted and 0 otherwise. Thus,  $(x_{kj})$  is a ‘solution’ and  $z_k := \sum_{j \in N_k} p_{kj} x_{kj}$  is the total profit for agent  $k$  with respect to this solution. (Throughout the paper,  $:=$  denotes definition, i.e., the notation on the left-hand side is defined by the right-hand side.) Except in Subsection 4.5, we limit the number of agents to  $K = 2$ , mainly for the sake of notational simplicity. We introduce the objective function of the form  $\phi(z_1, z_2)$  as a *social welfare function* that integrates individual preferences into an overall utility for all agents. Mathematically,  $\phi(\cdot, \cdot) : R_+^2 \rightarrow R_+$  is a continuous function satisfying the (rather mild) assumptions given in Section 2. Thus, we are concerned with the following *extended knapsack sharing problem*.

$$\text{XKSP : Maximize} \quad \phi\left(\sum_{j \in N_1} p_{1j} x_{1j}, \sum_{j \in N_2} p_{2j} x_{2j}\right) \quad (1.1)$$

$$\text{subject to} \quad \sum_{j \in N_1} w_{1j} x_{1j} + \sum_{j \in N_2} w_{2j} x_{2j} \leq C, \quad (1.2)$$

$$x_{kj} \in \{0, 1\}, \quad j \in N_k, \quad k = 1, 2. \quad (1.3)$$

This type of problem arises frequently when distributing a fixed amount of resources (money, manpower, time, etc.) over two or more agents with respective plans of projects. Each project requires a cost ( $w_{kj}$ ), and in return yields a profit ( $p_{kj}$ ) to the agent involved. For example, consider a newly married couple planning for their honeymoon. They are packing a suitcase with their respective items. Here,  $z_1$  and  $z_2$  are, respectively, their individual degrees of satisfaction with their packed items, and the overall degree of happiness for the couple is given by  $\phi(z_1, z_2)$ .

The *knapsack sharing problem* (KSP), originally formulated in [17, 18], is a special case of XKSP with  $\phi(z_1, z_2) = \min\{z_1, z_2\}$ , where the objective is to maximize the minimum profits of two agents. Heuristic and exact algorithms have been given for KSP in these and subsequent papers [1, 7]. Depending on the form of the welfare function  $\phi(z_1, z_2)$ , other special cases of XKSP include the following.

- *Multiplicative knapsack sharing problem* (MKSP): With  $\phi(z_1, z_2) = z_1 z_2$ , to maximize  $(\sum_{j \in N_1} p_{1j} x_{1j}) \cdot (\sum_{j \in N_2} p_{2j} x_{2j})$ ,
- *Quadratic knapsack sharing problem* (QKSP): With  $\phi(z_1, z_2) = (z_1)^2 + (z_2)^2$ , to maximize  $(\sum_{j \in N_1} p_{1j} x_{1j})^2 + (\sum_{j \in N_2} p_{2j} x_{2j})^2$ .

In this study, we are concerned with XKSP in general. In Section 2, we prove that XKSP is  $\mathcal{NP}$ -hard [6], and in Section 3, we present a dynamic programming (DP)-based pseudo-polynomial time algorithm to solve XKSP to optimality in a unified way. The existence of such an algorithm implies that XKSP is only *weakly*  $\mathcal{NP}$ -hard.

Since solving  $\mathcal{NP}$ -hard problems exactly in polynomial time in input data is believed to be essentially impossible unless  $\mathcal{P} = \mathcal{NP}$ , attempts have been made to solve these problems approximately in polynomial time under the PTAS (*polynomial time approximation scheme*) or FPTAS (*fully polynomial time approximation scheme*) [15, 16]. Various combinatorial optimization problems have been examined from this viewpoint. For example, Ibarra and Kim [8] developed an FPTAS for the *knapsack problem* [11, 14], and the time and space complexity of their algorithm were improved by Lawler [12] and subsequent researchers [9, 13]. FPTASs have also been given for the subset-sum problem [8, 10, 12] and its extensions [3–5].

Previously, approximation algorithms have been explored for the KSP mainly from a computational point of view [7, 18]. However, PTAS/FPTAS results have not been reported, i.e., from a theoretical viewpoint, these algorithms do not guarantee precision or polynomial run-time. In Section 4 of this paper, we present an FPTAS for XKSP with two agents ( $K = 2$ ). To construct such an FPTAS, we decompose XKSP into a series of subproblems and transform each of the subproblems into a two-dimensional optimization problem. Then, we apply *scaling* to compute the *weight functions* (introduced in Subsection 3.1) for these problems, and use the results to derive an FPTAS for XKSP. In the last part of Section 4, the FPTAS constructed for  $K = 2$  is extended to the case of more than two agents; and the developed algorithm gives an FPTAS for the problem with an externally given, fixed  $K (> 2)$ .

Section 5 concludes the paper, with perspectives on future research on this topic. In the Appendix, we give a non-DP-based PTAS for a limited case of the KSP with  $K = 2$ .

## 2. Assumptions and $\mathcal{NP}$ -hardness

We make the following assumptions throughout the paper, unless otherwise stated.

*Assumptions:*

A<sub>1</sub>. Problem data  $n_k, w_{kj}, p_{kj}$ , and  $C$  are all positive integers.

A<sub>2</sub>.  $w_{kj} \leq C$  for all  $j \in N_k, k = 1, 2$ .

A<sub>3</sub>.  $\min_{j \in N_1} \{w_{1j}\} + \min_{j \in N_2} \{w_{2j}\} \leq C$ .

A<sub>4</sub>.  $N_k$  is numbered in non-increasing order of  $p_{kj}$ , i.e.,

$$p_{k1} \geq p_{k2} \geq \cdots \geq p_{kn_k}, k = 1, 2.$$

A<sub>5</sub>.  $\phi(z_1, z_2)$  is *coordinate-wise non-decreasing*, i.e., for all  $(z_1, z_2) \in R_+^2$

$$z'_1 \leq z''_1 \Rightarrow \phi(z'_1, z_2) \leq \phi(z''_1, z_2) \text{ and } z'_2 \leq z''_2 \Rightarrow \phi(z_1, z'_2) \leq \phi(z_1, z''_2).$$

A<sub>6</sub>.  $\phi(0, z_2) \equiv \phi(z_1, 0) \equiv 0, \forall z_k \in R_+, k = 1, 2$ .

A<sub>7</sub>.  $\phi(z_1, z_2)$  can be computed in  $O(1)$  time.

A<sub>8</sub>.  $\phi(z_1, z_2)$  is a *super-homogeneous function* of degree  $r, (r \geq 1)$ , i.e.,

$$\phi(\lambda z_1, \lambda z_2) \geq \lambda^r \phi(z_1, z_2), \text{ for all } \lambda \geq 0 \text{ and } (z_1, z_2) \in R_+^2.$$

We introduce A<sub>2</sub> and A<sub>3</sub> to eliminate trivial cases. Without a loss of generality, we assume A<sub>4</sub> for the simplicity of subsequent descriptions of the algorithms. A<sub>5</sub> and A<sub>6</sub> are natural requirements for a social welfare function. Furthermore, we need A<sub>7</sub> and A<sub>8</sub> to derive FPTAS for XKSP in later sections.

KSP has been shown to be  $\mathcal{NP}$ -hard [18], as a generalization of the standard knapsack problem. Here we prove XKSP to be  $\mathcal{NP}$ -hard for an arbitrary welfare function  $\phi(z_1, z_2)$  satisfying the assumption A<sub>5</sub>.

**Theorem 2.1.** *XKSP is  $\mathcal{NP}$ -hard.*

*Proof:* Corresponding to a knapsack problem (KP) consisting of  $n$  items, with  $p_j$  and  $w_j$  representing the profit and weight of item  $j$ , respectively, and  $C$  indicating the knapsack capacity, consider an instance of XKSP where agent 1 has  $n$  items, agent 2 has only one item, and the capacity is  $C + 1$ . In this case, the items of agent 1 are all inherited from the KP, i.e., we have  $p_{1j} = p_j$  and  $w_{1j} = w_j$  for item  $j$ , while for agent 2 we set  $p_{21} = M$  and  $w_{21} = 1$ , where  $M > 0$  is some fixed number. In this instance, the item of agent 2 must be accepted, and thus the profit of agent 2 is  $M$ . Hence, the aim of the XKSP is to maximize  $\phi(\sum_{j=1}^n p_{1j}x_{1j}, M)$  subject to  $\sum_{j=1}^n w_{1j}x_{1j} \leq C$ . Due to the assumption A<sub>5</sub>, this is equivalent to the classic knapsack problem, which is known to be  $\mathcal{NP}$ -hard [14].  $\square$

### 3. DP-based Algorithm to Solve XKSP

In this section, we present a DP-based algorithm to solve small instances of XKSP to optimality. To construct such an algorithm, we transform the original problem to a two-dimensional optimization problem by introducing a *weight function*, as discussed below.

#### 3.1. Weight function

Let  $\bar{U}_k$  be an upper bound of  $\sum_{j \in N_k} p_{kj}x_{kj}$ , e.g.,

$$\bar{U}_k := \sum_{j \in N_k} p_{kj}. \tag{3.1}$$

For an arbitrary  $z_k \in [0, \bar{U}_k]_{int}$ , we introduce the following *inverse knapsack problem*, where  $[a, b]_{int}$  denotes the set of integers within interval  $[a, b]$ .

$$\begin{aligned} IKP_k(z_k) : \quad & \text{Minimize} && \sum_{j \in N_k} w_{kj}x_{kj} \\ & \text{subject to} && \sum_{j \in N_k} p_{kj}x_{kj} \geq z_k, \quad x_{kj} \in \{0, 1\}, \quad j \in N_k. \end{aligned}$$

The optimal objective value for this problem is denoted  $w_k^*(z_k)$ , and for the case in which the problem is infeasible, we set  $w_k^*(z_k) := \infty$ . Seen as a function of  $z_k$ ,  $w_k^*(z_k) : [0, \bar{U}_k]_{int} \rightarrow R_+ \cup \{\infty\}$  is referred to as the *weight function* of agent  $k$ . This can be computed by the following DP [2] algorithm. We define

$$W_k(i, z_k) := \min \left\{ \sum_{j=i+1}^{n_k} w_{kj} x_{kj} \mid \sum_{j=i+1}^{n_k} p_{kj} x_{kj} \geq z_k, x_{kj} \in \{0, 1\}, j \in \{i+1, \dots, n_k\} \right\}. \quad (3.2)$$

This gives the minimum weight for agent  $k$  when the total profit is not less than  $z_k$  and the first  $i$  items are not considered.

Following the DP *principle of optimality*, we derive the following recurrence relation. Starting with

$$W_k(n_k, z_k) = \begin{cases} \infty, & z_k > 0, \\ 0, & z_k = 0, \end{cases} \quad (3.3)$$

do the following (for each  $z_k \in [0, \bar{U}_k]_{int}$ ) from  $i = n_k - 1$  down to  $i = 0$ .

$$W_k(i, z_k) = \begin{cases} \min\{w_{ki} + W_k(i+1, z_k - p_{ki}), W_k(i+1, z_k)\}, & z_k \geq p_{ki}, \\ W_k(i+1, z_k), & z_k < p_{ki}, \end{cases} \quad (3.4)$$

$$x_k(i, z_k) = \begin{cases} 1, & z_k \geq p_{ki} \text{ and } w_{ki} + W_k(i+1, z_k - p_{ki}) \geq W_k(i+1, z_k), \\ 0, & \text{otherwise.} \end{cases} \quad (3.5)$$

The weight function for  $N_k$  is given as  $w_k^*(z_k) := W_k(0, z_k)$ . It is known that  $w_k^*(\cdot)$  is a non-decreasing, right-continuous step function of  $z_k \geq 0$  [11, 14].

### 3.2. Pseudo-polynomial time algorithm for XKSP

Once weight functions  $w_k^*(z_k)$  are obtained for  $k = 1, 2$ , we introduce the following problem.

$$\begin{aligned} \text{XKSP}' : \quad & \text{Maximize} && \phi(z_1, z_2) \\ & \text{subject to} && w_1^*(z_1) + w_2^*(z_2) \leq C, \\ & && z_k \in [0, \bar{U}_k]_{int}, \quad k = 1, 2. \end{aligned}$$

Let  $(z_1^*, z_2^*)$  denote an optimal solution to XKSP' with the objective value  $z^* := \phi(z_1^*, z_2^*)$ , and let  $\mathbf{x}_k^* = (x_{kj}^*)$  be an optimal solution to  $IKP_k(z_k^*)$ ,  $k = 1, 2$ . We set  $w_k^* := w_k^*(z_k^*) (= \sum_{j \in N_k} w_{kj} x_{kj}^*)$ . Then, by definition

$$w_1^* + w_2^* \leq C. \quad (3.6)$$

Additionally, by  $(\mathbf{x}_1^\circ, \mathbf{x}_2^\circ)$  we denote an optimal solution to XKSP, and we set  $w_k^\circ := \sum_{j \in N_k} w_{kj} x_{kj}^\circ$ ,  $z_k^\circ := \sum_{j \in N_k} p_{kj} x_{kj}^\circ$ , and  $z^\circ := \phi(z_1^\circ, z_2^\circ)$ . Again, we obtain

$$w_1^\circ + w_2^\circ \leq C, \quad (3.7)$$

and the following establishes the equivalence of XKSP and XKSP'.

**Theorem 3.1.** *Any optimal solution of XKSP is optimal for XKSP' and vice versa, i.e.,*

- (i)  $z^\circ = z^*$ ,
- (ii)  $(z_1^\circ, z_2^\circ)$  is an optimal solution to XKSP' with  $\mathbf{x}_k^\circ$  feasible for  $IKP_k(z_k^\circ)$ ,
- (iii)  $(\mathbf{x}_1^*, \mathbf{x}_2^*)$  is optimal to XKSP.

*Proof:* (Proof of  $z^\circ \leq z^*$ ) Since  $\mathbf{x}_k^\circ$  is a feasible solution to  $IKP_k(z_k^\circ)$ , we have  $w_k^*(z_k^\circ) \leq w_k^\circ$ . From (3.7),  $w_1^*(z_1^\circ) + w_2^*(z_2^\circ) \leq C$ , which implies that  $(z_1^\circ, z_2^\circ)$  is a feasible solution of  $XKSP'$ . Thus,  $z^\circ = \phi(z_1^\circ, z_2^\circ) \leq z^*$ .

(Proof of  $z^\circ \geq z^*$ ) From (3.6),  $(\mathbf{x}_1^*, \mathbf{x}_2^*)$  is a feasible solution to  $XKSP$ . This implies that  $z^\circ \geq \phi(\sum_{j \in N_1} p_{1j} x_{1j}^*, \sum_{j \in N_2} p_{2j} x_{2j}^*)$ . Furthermore, from the constraint of  $IKP_k(z_k^*)$ , we have  $\sum_{j \in N_k} p_{kj} x_{kj}^* \geq z_k^*$ . From this inequality and assumption  $A_5$ , we obtain  $\phi(\sum_{j \in N_1} p_{1j} x_{1j}^*, \sum_{j \in N_2} p_{2j} x_{2j}^*) \geq \phi(z_1^*, z_2^*) = z^*$ . Thus, (i) is proved.

So far, we have proved

$$z^\circ = \phi(z_1^\circ, z_2^\circ) = \phi\left(\sum_{j \in N_1} p_{1j} x_{1j}^*, \sum_{j \in N_2} p_{2j} x_{2j}^*\right) = \phi(z_1^*, z_2^*) = z^*,$$

which completes the proof of (ii) and (iii), as well.  $\square$

Thus,  $XKSP$  is solved by the DP-based algorithm `Solve_XKSP_DP` shown overleaf, which actually solves  $XKSP'$ .

### 3.3. Complexity

Given  $\bar{U}_k, (k = 1, 2)$  by equation (3.1), the computational complexity of the algorithm `Solve_XKSP_DP` is  $O(n_1 \bar{U}_1 + n_2 \bar{U}_2)$  for Step 1 to compute  $W_k(i, z_k), k = 1, 2$ , and thanks to the assumption  $A_7, O(\bar{U}_1 \bar{U}_2)$  to find the optimal  $(z_1^*, z_2^*)$  in Step 2. In total, the algorithm runs in pseudo-polynomial  $O(n_1 \bar{U}_1 + n_2 \bar{U}_2 + \bar{U}_1 \bar{U}_2)$  time. The memory requirement to keep  $W_k(i, z_k)$  for  $i = 1, \dots, n_k$  and  $z_k \in [0, \bar{U}_k]_{int}$  is  $O(n_1 \bar{U}_1 + n_2 \bar{U}_2)$ . This memory space is also sufficient for computing  $\mathbf{x}_1^*$  and  $\mathbf{x}_2^*$  in Step 3. Due to the existence of this pseudo-polynomial time algorithm,  $XKSP$  is only weakly  $\mathcal{NP}$ -hard [6].

## 4. FPTAS for the Problem $XKSP$

The DP-based algorithm described in the previous section solves  $XKSP$  to optimality in pseudo-polynomial time. In this section, we explore an approximation algorithm that runs in polynomial time in the size of the input data. For the optimal objective value  $z^*$  of  $XKSP$  and an arbitrary  $\epsilon \in [0, 1]$ , an algorithm is said to be a  $(1 - \epsilon)$ -factor approximation algorithm if its output  $z^\dagger$  satisfies  $z^\dagger \geq (1 - \epsilon)z^*$  for any instance of  $XKSP$ . If this algorithm runs, for a fixed  $\epsilon$ , in polynomial time in the size of the input data, this is said to be a *polynomial time approximation scheme* (PTAS). In addition, if the running time is polynomial with respect to  $1/\epsilon$  as well, this is said to be a *fully polynomial time approximation scheme* (FPTAS) [15, 16].

### 4.1. Subproblem

To construct such an approximation scheme for  $XKSP$ , we decompose the problem into subproblems  $P[i_1, i_2]$  ( $i_1 \in N_1, i_2 \in N_2$ ) by fixing the first  $i_k - 1$  of  $x_{kj}$ 's to 0 and  $x_{ki_k}$  to 1 for  $k = 1, 2$ , i.e.,

$$\begin{aligned} P[i_1, i_2] : \quad & \text{Maximize} && \phi\left(\sum_{j \in N_1} p_{1j} x_{1j}, \sum_{j \in N_2} p_{2j} x_{2j}\right) \\ & \text{subject to} && (1.2), (1.3), \\ & && x_{k1} = \dots = x_{ki_k-1} = 0, x_{ki_k} = 1; \quad k = 1, 2. \end{aligned} \quad (4.1)$$

**Algorithm** Solve\_XKSP\_DP

*Input:* Item data  $\{n_k, (w_{k,j}, p_{k,j}), j = 1, \dots, n_k\}$  for  $k = 1, 2$ , and the capacity  $C$ .  
*Output:* Optimal solution  $(\mathbf{x}_1^*, \mathbf{x}_2^*)$  and the corresponding objective value  $z^*$ .  
// Step 0. Preparation: Compute  $\bar{U}_k$ , ( $k = 1, 2$ ) by equation (3.1).  
// Step 1. Solve  $IKP_k(z_k)$  to obtain the weight functions  $w_k^*(z_k)$ ,  $k = 1, 2$ .  
// Initialization:  $i = n_k$   
**for**  $k = 1$  to  $2$  **do**  
  **for**  $z_k = 0$  to  $\bar{U}_k$  **do**  
    Initialize  $W_k(n_k, z_k)$  by equation (3.3).  
  **endfor**  
**endfor**  
// Backward recursion.  
**for**  $k = 1$  to  $2$  **do**  
  **for**  $i = n_k - 1$  down to  $0$  **do**  
    **for**  $z_k = 0$  to  $\bar{U}_k$  **do**  
      Compute  $W_k(i, z_k)$  and  $x_k(i, z_k)$  by equations (3.4) and (3.5), resp.  
    **endfor**  
  **endfor**  
  **for**  $z_k = 0$  to  $\bar{U}_k$  **do**  
     $w_k^*(z_k) \leftarrow W_k(0, z_k)$ .  
  **endfor**  
**endfor**  
// Step 2. Find the optimal solution  $(z_1^*, z_2^*)$  of XKSP'.  
set  $z^* = z_1^* = z_2^* = -\infty$ .  
**for**  $z_1 = 0$  to  $\bar{U}_1$  **do**  
  **for**  $z_2 = 0$  to  $\bar{U}_2$  **do**  
    **if**  $w_1^*(z_1) + w_2^*(z_2) > C$  **continue**.  
    **if**  $\phi(z_1, z_2) > z^*$  **do**  
       $z^* \leftarrow \phi(z_1, z_2), z_1^* \leftarrow z_1, z_2^* \leftarrow z_2$ .  
    **endif**  
  **endfor**  
**endfor**  
// Step 3. Retrieve the optimal solution and output.  
 $\mathbf{x}_1^*, \mathbf{x}_2^* \leftarrow$  solutions of  $IKP_1(z_1^*)$  and  $IKP_2(z_2^*)$ .  
Output the optimal  $(\mathbf{x}_1^*, \mathbf{x}_2^*)$  with the objective value  $z^*$ , and **stop**.

Clearly, XKSP is solved by solving all of these subproblems. Corresponding to these subproblems, we introduce the following inverse knapsack problem for agent  $k$  with the first  $i_k - 1$  variables fixed to 0, and the  $i_k$ -th to 1.

$$\begin{aligned}
IKP_k(z_k | i_k) : \quad & \text{Minimize} && \sum_{j \in N_k} w_{kj} x_{kj} \\
& \text{subject to} && \sum_{j \in N_k} p_{kj} x_{kj} \geq z_k, \quad x_{kj} \in \{0, 1\}, j \in N_k, \\
& && x_{k1} = \dots = x_{ki_k-1} = 0, x_{ki_k} = 1.
\end{aligned} \tag{4.2}$$

$w_k^*(z_k|i_k)$  denotes the optimal objective value of this problem. If this problem is infeasible, we set  $w_k^*(z_k|i_k) = \infty$ .

Then, as in Theorem 3.1,  $P[i_1, i_2]$  can be rewritten as

$$\begin{aligned} P'[i_1, i_2] : \quad & \text{Maximize} && \phi(z_1, z_2) \\ & \text{subject to} && w_1^*(z_1|i_1) + w_2^*(z_2|i_2) \leq C, \\ & && z_k \in [0, \bar{U}_k[i_k]]_{int}, k = 1, 2, \end{aligned}$$

where  $\bar{U}_k[i_k]$  is an upper bound of  $\sum_{j \in N_k} p_{kj} x_{kj}$  under  $x_{k1} = \dots = x_{ki_k-1} = 0, x_{ki_k} = 1$ . With this constraint and assumption  $A_4$ , we obtain the following:

$$\bar{U}_k[i_k] = (n_k - i_k + 1)p_{ki_k}. \quad (4.3)$$

Thus,  $P[i_1, i_2]$  is solved by solving  $P'[i_1, i_2]$ . Let  $(z_1^*[i_1, i_2], z_2^*[i_1, i_2])$  be an optimal solution to  $P'[i_1, i_2]$ , and  $\mathbf{x}_k^*[i_1, i_2] = (x_{kj}^*[i_1, i_2])$  denotes an optimal solution to  $IKP_k(z_k^*[i_1, i_2]|i_k)$ . Here, we note

$$p_{ki_k} \leq z_k^*[i_1, i_2] \leq \bar{U}_k[i_k], \quad (4.4)$$

which follows directly from assumption  $A_4$  and  $x_{ki_k} = 1$  for this subproblem.

#### 4.2. Scaling of $IKP_k(z_k|i_k)$

For an arbitrary  $\epsilon \in [0, 1]$ , let

$$H_k[i_k] := \epsilon p_{ki_k} / (n_k - i_k + 1) \quad (4.5)$$

be the scaling factor to divide the range  $[0, \bar{U}_k[i_k]]$  of  $z_k^*[i_1, i_2]$  into

$$\bar{V}_k[i_k] := \lfloor (n_k - i_k + 1)^2 / \epsilon \rfloor \quad (4.6)$$

sub-intervals of width  $H_k[i_k]$ . We introduce the scaling of profits

$$q_{kj}[i_k] = \lfloor p_{kj} / H_k[i_k] \rfloor \quad (4.7)$$

and the scaled subproblem

$$\begin{aligned} Q[i_1, i_2] : \quad & \text{Maximize} && \phi(H_1[i_1] \sum_{j \in N_1} q_{1j}[i_1] x_{1j}, H_2[i_2] \sum_{j \in N_2} q_{2j}[i_2] x_{2j}) \\ & \text{subject to} && (1.2), (1.3), (4.1). \end{aligned}$$

Further, we introduce the inverse problem for  $z_k \in [0, \bar{V}_k[i_k]]_{int}$  as

$$\begin{aligned} IKQ_k(z_k|i_k) : \quad & \text{Minimize} && \sum_{j \in N_k} w_{kj} x_{kj} \\ & \text{subject to} && \sum_{j \in N_k} q_{kj}[i_k] x_{kj} \geq z_k, \quad x_{kj} \in \{0, 1\}, j \in N_k, \\ & && (4.2). \end{aligned}$$

Let  $w_k^\dagger(z_k|i_k)$  be the optimal objective value for this problem. This can be computed, analogously to the weight function  $w_k^*(z_k)$  introduced in Section 3, by the DP recursion procedure. Then, again as in Theorem 3.1,  $Q[i_1, i_2]$  can be rewritten as

$$\begin{aligned} Q'[i_1, i_2] : \quad & \text{Maximize} && \phi(H_1[i_1]z_1, H_2[i_2]z_2) \\ & \text{subject to} && w_1^\dagger(z_1|i_1) + w_2^\dagger(z_2|i_2) \leq C, \\ & && z_k \in [0, \bar{V}_k[i_k]]_{int}, \quad k = 1, 2. \end{aligned}$$

By  $(z_1^\dagger[i_1, i_2], z_2^\dagger[i_1, i_2])$ , we denote an optimal solution to this problem, and  $\mathbf{x}_k^\dagger[i_1, i_2] = (x_{kj}^\dagger[i_1, i_2])$  is an optimal solution to  $IKQ_k(z_k^\dagger[i_1, i_2]|i_k)$ . Again,  $(\mathbf{x}_1^\dagger[i_1, i_2], \mathbf{x}_2^\dagger[i_1, i_2])$  gives an optimal solution to  $Q[i_1, i_2]$ .

### 4.3. Approximation algorithm for XKSP

The solution  $(\mathbf{x}_1^\dagger[i_1, i_2], \mathbf{x}_2^\dagger[i_1, i_2])$  of  $Q[i_1, i_2]$ , as obtained above, is a feasible solution to  $P[i_1, i_2]$ , which we employ as an approximate solution with the objective value

$$z^\dagger[i_1, i_2] = \phi\left(\sum_{j \in N_1} p_{1j} x_{1j}^\dagger[i_1, i_2], \sum_{j \in N_2} p_{2j} x_{2j}^\dagger[i_1, i_2]\right).$$

The following algorithm `Solve_XKSP_Approx`, shown overleaf, solves XKSP approximately by finding the best among all  $z^\dagger[i_1, i_2]$ 's, i.e., with  $(i_1^\dagger, i_2^\dagger)$  as a maximizer, we have

$$z^\dagger = z^\dagger[i_1^\dagger, i_2^\dagger] = \max\{z^\dagger[i_1, i_2] \mid i_1 \in N_1, i_2 \in N_2\}. \quad (4.8)$$

The computational complexity of this algorithm is analyzed below. In solving  $IKQ_k(z_k|i_k)$  to compute  $w_k^*(z_k|i_k)$  for  $z_k \in [0, \bar{V}_k[i_k]]_{int}$  and  $i = 1, 2, \dots, n_k$ , the DP algorithm runs in  $(n_k - i_k + 1)\bar{V}_k[i_k] = O((n_k - i_k + 1)^3/\epsilon)$  time. In total, Step 1 needs  $\sum_{1 \leq i_1 \leq n_1} (n_1 - i_1 + 1)^3/\epsilon + \sum_{1 \leq i_2 \leq n_2} (n_2 - i_2 + 1)^3/\epsilon = O((n_1^4 + n_2^4)/\epsilon)$  time. To find the optimal  $z^\dagger[i_1, i_2]$  in Step 2, again due to the assumption  $A_7$ ,  $\bar{V}_1[i_1]\bar{V}_2[i_2] = O((n_1 - i_1 + 1)^2(n_2 - i_2 + 1)^2/\epsilon^2)$  time suffices. Thus, searching for the optimal  $z^\dagger$  requires  $\sum_{1 \leq i_1 \leq n_1} \sum_{1 \leq i_2 \leq n_2} (n_1 - i_1 + 1)^2(n_2 - i_2 + 1)^2/\epsilon^2 = O(n_1^3 n_2^3/\epsilon^2)$  time. For Step 3,  $(n_1 - i_1^\dagger + 1)\bar{V}_1[i_1^\dagger] + (n_2 - i_2^\dagger + 1)\bar{V}_2[i_2^\dagger] = O((n_1^3 + n_2^3)/\epsilon)$  time suffices, which is negligible with respect to  $O((n_1^4 + n_2^4)/\epsilon)$  time for Step 1. In summary, the total time complexity of this algorithm is  $O((n_1^4 + n_2^4)/\epsilon + n_1^3 n_2^3/\epsilon^2)$ .

The space requirement for the DP procedure to solve  $IKQ_k(z_k|i_k)$  is at most  $(n_k)\bar{V}_k[1] = O(n_k^3/\epsilon)$ . This space is reusable for  $i_k = 1, 2, \dots, n_k$ , and is sufficient to keep  $w_k^*(z_k|i_k)$  in memory as well. Thus, for Step 1, we need  $O((n_1^3 + n_2^3)/\epsilon)$  space. In Step 2, we note that  $z^\dagger[i_1, i_2]$  is used only within the loop of  $(i_1, i_2)$ . This means that it may be replaced, e.g., by  $z^\ddagger$ , and  $z^\ddagger$  can be used repeatedly for all pairs of  $(i_1, i_2) \in N_1 \times N_2$ . This is true for  $z_1^\dagger[i_1, i_2]$  and  $z_2^\dagger[i_1, i_2]$ ; thus, the memory needed for Step 2 is only  $O(1)$ . For Step 3, the space used for Step 1 is again reusable. Therefore, the total space requirement of `Solve_XKSP_Approx` is  $O((n_1^3 + n_2^3)/\epsilon)$ .

### 4.4. Proof of FPTAS of the approximation algorithm

We prove that `Solve_XKSP_Approx` is an  $(1 - \epsilon)$ -approximation algorithm for XKSP. First, we prepare the following.

**Lemma 4.1.**

$$H_k[i_k] \sum_{j \in N_k} q_{kj}[i_k] x_{kj}^*[i_1, i_2] \geq (1 - \epsilon) z_k^*[i_1, i_2], \quad k = 1, 2.$$

**Algorithm** Solve\_XKSP\_Approx

*Input:* Item data  $\{n_k, (w_{k,j}, p_{k,j}), j = 1, \dots, n_k\}$  for  $k = 1, 2$ , and the capacity  $C$ .  
*Output:* Approximate solution  $(\mathbf{x}_1^\dagger, \mathbf{x}_2^\dagger)$  and the corresponding objective value  $z^\dagger$ .  
 // Step 1. Scale & solve  $IKQ_k(z_k|i_k)$  to obtain  $w_k^\dagger(z_k|i_k)$ .  
 // Scaling  
**for**  $k = 1$  to  $2$  **do**  
   **for**  $i_k = 1$  to  $n_k$  **do**  
     Initialize  $H_k[i_k]$  and  $\bar{V}[i_k]$  by equations (4.5) and (4.6), resp.  
     **for**  $j = 1$  to  $n_k$  **do**  
       Initialize  $q_{kj}[i_k]$  by equation (4.7).  
     **endfor**  
     Compute  $w_k^\dagger(z_k|i_k)$  by solving  $IKQ_k(z_k|i_k)$  via the DP procedure.  
   **endfor**  
**endfor**  
 // Step 2. Compute the optimal  $z^\dagger$  and  $(i_1^\dagger, i_2^\dagger) = \arg \max\{z^\dagger[i_1, i_2]\}$ .  
 set  $i_1^\dagger = i_2^\dagger = 0, z^\dagger = z_1^\dagger = z_2^\dagger = -\infty$ .  
**for**  $i_1 = 1$  to  $n_1$  **do**  
   **for**  $i_2 = 1$  to  $n_2$  **do**  
      $z^\dagger[i_1, i_2] = z_1^\dagger[i_1, i_2] = z_2^\dagger[i_1, i_2] = -\infty$ .  
     **for**  $z_1 = 0$  to  $\bar{V}_1[i_1]$  **do**  
       **for**  $z_2 = 0$  to  $\bar{V}_2[i_2]$  **do**  
         **if**  $w_1^\dagger(z_1|i_1) + w_2^\dagger(z_2|i_2) > C$  **continue**.  
         **if**  $\phi(H_1[i_1]z_1, H_2[i_2]z_2) > z^\dagger[i_1, i_2]$  **do**  
            $z^\dagger[i_1, i_2] \leftarrow \phi(H_1[i_1]z_1, H_2[i_2]z_2),$   
            $z_1^\dagger[i_1, i_2] = z_1, z_2^\dagger[i_1, i_2] = z_2.$   
         **endif**  
       **endfor**  
     **endfor**  
     **if**  $z^\dagger[i_1, i_2] > z^\dagger$  **do**  
        $i_1^\dagger \leftarrow i_1, i_2^\dagger \leftarrow i_2, z^\dagger \leftarrow z^\dagger[i_1, i_2], z_1^\dagger \leftarrow z_1^\dagger[i_1, i_2], z_2^\dagger \leftarrow z_2^\dagger[i_1, i_2].$   
     **endif**  
   **endfor**  
**endfor**  
 // Step 3. Retrieve the optimal solution and output.  
 $\mathbf{x}_1^\dagger, \mathbf{x}_2^\dagger \leftarrow$  solutions of  $IKQ_1(z_1^\dagger|i_1^\dagger)$  and  $IKQ_2(z_2^\dagger|i_2^\dagger)$ .  
 Output the approximate solution  $(\mathbf{x}_1^\dagger, \mathbf{x}_2^\dagger)$  with the objective value  $z^\dagger$ , and **stop**.

*Proof:*

$$\begin{aligned}
 & H_k[i_k] \sum_{j \in N_k} q_{kj}[i_k] x_{kj}^*[i_1, i_2] = H_k[i_k] \sum_{j \in N_k} [p_{kj}/H_k[i_k]] x_{kj}^*[i_1, i_2] \\
 & \geq H_k[i_k] \sum_{j \in N_k} (p_{kj}/H_k[i_k] - 1) x_{kj}^*[i_1, i_2] \\
 & \geq \sum_{j \in N_k} p_{kj} x_{kj}^*[i_1, i_2] - H_k[i_k](n_k - i_k + 1) \\
 & \geq z_k^*[i_1, i_2] - \epsilon p_{ki_k} \geq (1 - \epsilon) z_k^*[i_1, i_2],
 \end{aligned}$$

where the last inequality follows from the constraint

$$\sum_{j \in N_k} p_{kj} x_{kj}^*[i_1, i_2] \geq z_k^*[i_1, i_2]$$

for  $IKP_k(i_k | z_k^*[i_1, i_2])$  and (4.4). □

**Proposition 4.1.**

$z^\dagger[i_1, i_2] \geq (1 - r\epsilon)z^*[i_1, i_2]$  for all  $(i_1, i_2) \in N_1 \times N_2$ .

*Proof:*

$$\begin{aligned} z^\dagger[i_1, i_2] &= \phi\left(\sum_{j \in N_1} p_{1j} x_{1j}^\dagger[i_1, i_2], \sum_{j \in N_2} p_{2j} x_{2j}^\dagger[i_1, i_2]\right) \\ &\geq \phi\left(H_1[i_1] \sum_{j \in N_1} q_{1j}[i_1] x_{1j}^\dagger[i_1, i_2], H_2[i_2] \sum_{j \in N_2} q_{2j}[i_1] x_{2j}^\dagger[i_1, i_2]\right) \\ &\quad (\text{from } p_{kj} \geq H_k[i_k] q_{kj}[i_k] \text{ and Assumption } A_5) \\ &\geq \phi\left(H_1[i_1] z_1^\dagger[i_1, i_2], H_2[i_2] z_2^\dagger[i_1, i_2]\right) \\ &\quad (\text{from } \sum_{j \in N_k} q_{kj}[k_j] x_{kj}^\dagger[i_1, i_2] \geq z_k^\dagger[i_1, i_2] \text{ and Assumption } A_5) \\ &\geq \phi\left(H_1[i_1] \sum_{j \in N_1} q_{1j}[i_1] x_{1j}^*[i_1, i_2], H_2[i_2] \sum_{j \in N_2} q_{2j}[i_1] x_{2j}^*[i_1, i_2]\right) \\ &\quad (\text{since } (z_1^\dagger[i_1, i_2], z_2^\dagger[i_1, i_2]) \text{ is optimal for } Q'[i_1, i_2]) \\ &\geq \phi\left((1 - \epsilon)z_1^*[i_1, i_2], (1 - \epsilon)z_2^*[i_1, i_2]\right) \quad (\text{from Lemma 4.1}) \\ &\geq (1 - \epsilon)^r \phi\left(z_1^*[i_1, i_2], z_2^*[i_1, i_2]\right) \quad (\text{from Assumption } A_8) \\ &\geq (1 - r\epsilon)z^*[i_1, i_2]. \quad \square \end{aligned}$$

Analogous to (4.8), the optimal objective value  $z^*$  is the best on out of all  $z^*[i_1, i_2]$ 's; with  $(i_1^*, i_2^*)$  as a maximizer, this yields

$$z^* = z^*[i_1^*, i_2^*] = \max\{z^*[i_1, i_2] | i_1 \in N_1, i_2 \in N_2\}. \quad (4.9)$$

Then, we obtain the following.

**Theorem 4.1.** *Solve\_XKSP\_Approx gives an FPTAS for XKSP.*

*Proof:* We already proved that *Solve\_XKSP\_Approx* runs in polynomial time of  $n_1, n_2$  and  $1/\epsilon$ . This combined with  $z^\dagger = z^\dagger[i_1^\dagger, i_2^\dagger] \geq z^\dagger[i_1^*, i_2^*] \geq (1 - r\epsilon)z^*[i_1^*, i_2^*] = (1 - r\epsilon)z^*$ , which follows from Proposition 4.1, proves Theorem 4.1. □

**4.5. The case of  $K > 2$**

Although we focused on the case of two agents ( $K = 2$ ), most of the results can be extended to the case of  $K > 2$  agents, provided that  $K$  is an externally given, fixed number. XKSP with  $K \geq 2$  is solved by solving

$$\begin{aligned} \text{XKSP}' : \quad &\text{Maximize} && \phi(z_1, z_2, \dots, z_K) \\ &\text{subject to} && w_1^*(z_1) + w_2^*(z_2) + \dots + w_K^*(z_K) \leq C, \\ & && 0 \leq z_k \leq \bar{U}_k, \quad k = 1, 2, \dots, K. \end{aligned}$$

This can be solved in  $O(n_1 \bar{U}_1 + n_2 \bar{U}_2 + \dots + n_K \bar{U}_K)$  time to compute weight functions  $w_k^*(z_k)$  in  $O(\bar{U}_1 \bar{U}_2 \dots \bar{U}_K)$  time to find the optimal  $(z_1^*, z_2^*, \dots, z_K^*)$  that maximizes  $\phi(z_1, z_2, \dots, z_K)$ .

Thus, we have a  $O(n_1\bar{U}_1 + n_2\bar{U}_2 + \dots + n_K\bar{U}_K + \bar{U}_1\bar{U}_2 \dots \bar{U}_K)$  time pseudo-polynomial time algorithm.

Solve\_XKSP\_Approx can be extended to the  $K > 2$  case, as well. We decompose the problem into subproblems  $P[i_1, i_2, \dots, i_K]$ , and through (4.5)-(4.7) introduce the scaled subproblem  $Q[i_1, i_2, \dots, i_K]$ . Weight functions  $w_k^\dagger(z|i_k)$  can be computed in  $O((n_1^4 + n_2^4 + \dots + n_K^4)/\epsilon)$  time, and the approximate value  $z^\dagger[i_1, i_2, \dots, i_K]$  is obtained in  $\bar{U}_1[i_1]\bar{U}_2[i_2] \dots \bar{U}_K[i_K] = O((n_1^3/\epsilon)(n_2^3/\epsilon) \dots (n_K^3/\epsilon))$  time. Thus, in total, we have a FPTAS that runs in  $O((n_1^4 + n_2^4 + \dots + n_K^4)/\epsilon + n_1^3n_2^3 \dots n_K^3/\epsilon^K)$  time. Thus, the algorithm is an FPTAS for XKSP, provided that  $K$  is an externally given, fixed number.

## 5. Conclusion

In this article, we formulated the XKSP as a family of problems, including KSP, MKSP, and QKSP. We developed a DP-based algorithm to solve this family of problems to optimality in a unified way, and proved that this problem is  $\mathcal{NP}$ -hard in the weak sense. Based on the DP-based algorithm, we also constructed an algorithm to solve the problem approximately in polynomial time by decomposing it into a series of subproblems and by introducing scaling in the DP computation of the subproblems. The algorithm was proven to give an FPTAS to the XKSP. In most parts of the paper, the algorithm was described for the case of two agents ( $K = 2$ ), but it can be extended to the case of  $K > 2$ , if  $K$  is an externally given, fixed number. For KSP with  $K = 2$  agents, we shall give a simpler approximation algorithm, which is proven to be a PTAS, in the Appendix.

To our knowledge, this paper is the first to explore approximation algorithms for KSP and its extensions from a theoretical FPTAS/PTAS point of view. We highlight the following prospective future research issues.

- Improving the time/space complexity of Solve\_XKSP\_Approx. This may be possible by improvements in the FPTAS for KP based on [8, 9, 12], etc.
- Developing a PTAS for KSP with  $K > 2$ , or extending PTAS to XKSP in general, and at least to the multiplicative/quadratic KSP with  $K = 2$  or more.
- Developing an approximation algorithm that runs in polynomial time of  $K$  may be a worthwhile challenge.

**Acknowledgments** The authors would like to thank the anonymous referees and the Editor of this journal for their helpful remarks and suggestions on our manuscript.

## Appendix: PTAS for KSP with $K = 2$

We develop a greedy algorithm for the limited case of KSP with two agents and prove that it is a PTAS. Throughout the Appendix, assumptions  $A_1$ - $A_3$  are maintained, while  $A_4$  is replaced with the following.

$A_9$ :  $N_k$  is numbered in non-increasing order of the profit per weight, i.e.,

$$p_{k1}/w_{k1} \geq p_{k2}/w_{k2} \dots \geq p_{kn_k}/w_{kn_k}, k = 1, 2.$$

We consider a slightly extended problem

$$\begin{aligned} KSP(z_{10}, z_{20}, \bar{c}) : \quad & \text{Maximize} && \min(z_{10} + \sum_{j \in N_1} p_{1j}x_{1j}, z_{20} + \sum_{j \in N_2} p_{2j}x_{2j}) \\ & \text{subject to} && \sum_{j \in N_1} w_{1j}x_{1j} + \sum_{j \in N_2} w_{2j}x_{2j} \leq \bar{c}, x_{kj} \in \{0, 1\}, \end{aligned}$$

where  $z_{10}$  and  $z_{20}$  are ‘fixed’ scores given *a priori* to each group, and  $KSP(0, 0, C)$  is the original KSP.

We then introduce the knapsack problem

$$\begin{aligned} KP_k(z_{k0}, c) : \quad & \text{Maximize} && z_{k0} + \sum_{j \in N_k} p_{kj} x_{kj} \\ & \text{subject to} && \sum_{j \in N_k} w_{kj} x_{kj} \leq c, \quad x_{kj} \in \{0, 1\}. \end{aligned}$$

By  $\overline{KP}_k(z_{k0}, c)$ , we denote the continuous relaxation of this problem, where 0-1 constraint  $x_{kj} \in \{0, 1\}$  is replaced with  $0 \leq x_{kj} \leq 1$ . The optimal objective value of  $\overline{KP}_k(z_{k0}, c)$  is denoted  $\bar{z}_k(z_{k0}, c)$ , and  $\bar{w}_{kj} := \sum_{1 \leq i \leq j} w_{ki}$  ( $\bar{p}_{kj} := z_{k0} + \sum_{1 \leq i \leq j} p_{ki}$ , resp.) is the *accumulated* weight (value, resp.) of items in  $N_k$ . Due to  $A_9$ ,  $\bar{z}_k(z_{k0}, c)$  is a piecewise linear, concave, and monotonically increasing function of  $c$ , which is obtained by connecting points  $(\bar{w}_{k0}, \bar{p}_{k0}), (\bar{w}_{k1}, \bar{p}_{k1}), (\bar{w}_{k2}, \bar{p}_{k2}), \dots, (\bar{w}_{kn_k}, \bar{p}_{kn_k})$ .

If  $\bar{z}_1(z_{10}, \bar{c}) \geq z_{20}$  and  $\bar{z}_2(z_{20}, \bar{c}) \geq z_{10}$ , then  $\bar{z}_1(z_{10}, c) = \bar{z}_2(z_{20}, \bar{c} - c)$  has a unique solution, say  $\hat{c}$ , and the corresponding value  $\hat{z} := \bar{z}_1(z_{10}, \hat{c})$ . This gives an upper bound to  $KSP(z_{10}, z_{20}, \bar{c})$ , since this is the optimal objective value for the continuous relaxation of the problem. Thus, for the optimal objective value  $z^*(z_{10}, z_{20}, \bar{c})$  of  $KSP(z_{10}, z_{20}, \bar{c})$ , we have

$$z^*(z_{10}, z_{20}, \bar{c}) \leq \hat{z}.$$

Let  $s \in N_1$  be the item satisfying  $\bar{w}_{1,s-1} \leq \hat{c} < \bar{w}_{1s}$ . Such an item is said to be *critical* in  $N_1$ . Similarly, the critical item  $t$  in  $N_2$  satisfies  $\bar{w}_{2,t-1} \leq C - \hat{c} < \bar{w}_{2t}$ . Then,

$$\underline{z}(z_{10}, z_{20}, \bar{c}) := \min(\bar{p}_{1,s-1}, \bar{p}_{2,t-1})$$

gives a lower bound to  $KSP(z_{10}, z_{20}, \bar{c})$ , and  $S(z_{10}, z_{20}, \bar{c}) := \{j \in N_1 | j < s\} \cup \{j \in N_2 | j < t\}$  is a feasible solution to this problem. Furthermore, since  $\bar{p}_{1s} := \bar{p}_{1,s-1} + p_{1s} \geq \hat{z}$  and  $\bar{p}_{2t} := \bar{p}_{2,t-1} + p_{2t} \geq \hat{z}$ , both of these give upper bounds to  $KSP(z_{10}, z_{20}, \bar{c})$ . We take

$$\bar{z}(z_{10}, z_{20}, \bar{c}) := \begin{cases} \bar{p}_{1s}, & \text{if } \bar{p}_{1,s-1} \leq \bar{p}_{2,t-1}, \\ \bar{p}_{2t}, & \text{otherwise} \end{cases}$$

as the upper bound of  $KSP(z_{10}, z_{20}, \bar{c})$  to be used hereafter.

For an arbitrary subset  $F_k \subseteq N_k$  ( $k = 1, 2$ ), we write  $w_k(F_k) := \sum_{j \in F_k} w_{kj}$  and  $p_k(F_k) := \sum_{j \in F_k} p_{kj}$  to simplify the notation, and for a pair of subsets  $(F_1, F_2)$ , we introduce the following to solve KSP (approximately) with  $F_1 \cup F_2$  fixed a priori.

### Procedure GREEDY( $F_1, F_2$ )

- Put  $c_{k0} := w_k(F_k)$  and  $z_{k0} := p_k(F_k)$ ,  $k = 1, 2$ ,  $C' := C - c_{10} - c_{20}$ .
- With  $KSP(z_{10}, z_{20}, C')$ , compute a pair of critical items  $(s, t)$ , the lower bound  $\underline{z}(F_1, F_2) := \underline{z}(z_{10}, z_{20}, C')$ , and the corresponding feasible solution  $S(F_1, F_2) := S(z_{10}, z_{20}, C')$ .

The running time of this procedure is  $O(n_1 + n_2)$ .

We now describe a PTAS for KSP. Let  $\epsilon \in (0, 1)$  be a fixed constant and let  $h := \lceil 1/\epsilon \rceil$ . We first try to guess the most profitable set of items from each group and pack the rest greedily.

**Algorithm** Approx\_KSP

**for** each pair  $(F_1, F_2)$  satisfying  $F_k \subseteq N_k, |F_k| \leq h, k = 1, 2$  **do**  
 - Pack  $F_1$  and  $F_2$  into the knapsack.  
 - Remove from  $N_k \setminus F_k$  all items  $j$  satisfying  $p_{kj} > p_{ki}$ , where  
 $i := \arg \min\{p_{kj} | j \in F_k\}$  is the least profitable item in  $F_k$ .  
 - With the sets of remaining items, run GREEDY( $F_1, F_2$ ) and obtain  
 the lower bound  $\underline{z}(F_1, F_2)$ , together with the feasible solution  $S(F_1, F_2)$ .  
**endfor**  
 Output the best solution from above, and **stop**.

**Theorem A.1.** Approx\_KSP gives a PTAS for KSP with  $K = 2$ .

*Proof:* (Complexity) There are  $n_1 C_h \cdot n_2 C_h < n_1^h \cdot n_2^h$  pairs of sets satisfying  $|F_1| \leq h, |F_2| \leq h$ , and for each pair  $(F_1, F_2)$  we need  $O(n_1 + n_2)$  time. Accordingly, the total time complexity is  $O((n_1 + n_2)n_1^h n_2^h) = O((n_1 + n_2)(n_1 n_2)^{\lceil 1/\epsilon \rceil})$ .

(Accuracy) Let  $(O_1^*, O_2^*)$  be an optimal solution of the original  $KSP \equiv KSP(0, 0, C)$  with the optimal objective value  $z^* := \min\{p(O_1^*), p(O_2^*)\}$ . Here, for  $k = 1, 2$  we set  $F_k$  to the first  $h$  most profitable items in  $O_k^*$  if  $|O_k^*| \geq h$ , and we set  $F_k = O_k^*$  otherwise. Since  $F_k \subseteq O_k^*$ , KSP is solved by solving  $KSP(p_1(F_1), p_2(F_2), C - w_1(F_1) - w_2(F_2))$ , and thus  $z^*(p_1(F_1), p_2(F_2), C - w_1(F_1) - w_2(F_2)) = z^*$ . Let  $s$  and  $t$  be the critical items obtained by running GREEDY( $F_1, F_2$ ). Under this setting, we consider the following cases.

- (i) If  $|O_1^*| < h$  and  $\bar{p}_{2,t-1} \geq p(O_1^*)$ .  
 In this case,  $O_2^*$  can be regarded as an optimal solution to  $KP_2(0, C - w(F_1))$ , and we have  $p(O_2^*) \geq \bar{p}_{2,t-1}$ . Thus, we have  $p(O_2^*) \geq p(O_1^*)$ , which implies  $z^* = \min\{p(O_1^*), p(O_2^*)\} = p(O_1^*)$ . At the same time,  $\underline{z}(F_1, F_2) = \min\{p(F_1), \bar{p}_{2,t-1}\} = p(F_1) = p(O_1^*)$ . We thus conclude  $\underline{z}(F_1, F_2) = z^*$ .
- (ii) If  $|O_2^*| < h$  and  $\bar{p}_{1,s-1} \geq p(F_2)$ .  
 By symmetry, we have  $\underline{z}(F_1, F_2) = z^*$  as above.
- (iii) If  $|O_1^*| \geq h$  and  $\bar{p}_{1,s-1} \leq \bar{p}_{2,t-1}$ .  
 In this case, after running GREEDY( $F_1, F_2$ ), we have  $p_{1j} \geq p_{1s}$  for all  $j \in F_1$ . Then, if  $p_{1s} > \epsilon z^*$ , we obtain  $z^* \geq \underline{z}(F_1, F_2) = \bar{p}_{1,s-1} \geq \sum_{j \in F_1} p_{1j} \geq \sum_{j \in F_1} p_{1s} > h\epsilon z^* \geq z^*$ , which is a contradiction. Accordingly,  $p_{1s} \leq \epsilon z^*$  and from  $\bar{z}(z_{10}, z_{20}, C) = \underline{z}(F_1, F_2) + p_{1s} \geq z^*$ , we conclude  $\underline{z}(F_1, F_2) \geq (1 - \epsilon)z^*$ .
- (iv) If  $|F_2| \geq h$  and  $\bar{p}_{1,s-1} \leq \bar{p}_{2,t-1}$ .  
 Again, by symmetry to (iii) we have  $\underline{z}(F_1, F_2) \geq (1 - \epsilon)z^*$ .

Thus, in all cases,  $\underline{z}(F_1, F_2) \geq (1 - \epsilon)z^*$  holds, which completes the proof. □

**References**

- [1] T. Belgacem and M. Hifi: Sensitivity analysis of the knapsack sharing problem: Perturbation of the weight of an item. *Computers & Operations Research*, **35** (2008), 295–308.
- [2] R. Bellman: *Dynamic Programming* (Princeton University Press, Princeton, 1957).
- [3] A. Caprara, H. Kellerer, and U. Pferschy: The multiple subset sum problem. *SIAM Journal on Optimization*, **11** (2000), 308–319.
- [4] A. Caprara, H. Kellerer, U. Pferschy, and D. Pisinger: Approximation algorithms for

- knapsack problems with cardinality constraints. *European Journal of Operational Research*, **123** (2000), 333–345.
- [5] M. Dawande and J. Kalagnanam: Approximation algorithms for the multiple knapsack problem with assignment restrictions. *Journal of Combinatorial Optimization*, **4** (2000), 171–186.
- [6] M.R. Garey and D.S. Johnson: *Computers and Intractability: A Guide to the Theory of NP-Completeness* (Freeman and Company, San Francisco, CA, 1979).
- [7] M. Hifi, H. M’Halla, and S. Sadfi: An exact algorithm for the knapsack sharing problem. *Computers & Operations Research*, **32** (2005), 1311–1324.
- [8] O.H. Ibarra and C.E. Kim: Fast approximation algorithms for knapsack problem and sum of subset problems. *Journal of the Association for Computing Machinery*, **22** (1975), 463–468.
- [9] H. Kellerer and U. Pferschy: A new fully polynomial time approximation scheme for the knapsack problem. *Journal of Combinatorial Optimization*, **3** (1999), 59–71.
- [10] H. Kellerer, U. Pferschy, R. Mansini, and M.G. Speranza: An efficient fully polynomial time approximation scheme for the subset-sum problem. *Journal of Computer and Systems Sciences*, **66** (2003), 349–370.
- [11] H. Kellerer, U. Pferschy, and D. Pisinger: *Knapsack Problems* (Springer, Berlin, 2004).
- [12] E.L. Lawler: Fast approximation algorithms for knapsack problems. *Mathematics of Operations Research*, **4** (1979), 339–356.
- [13] M.J. Magazine and O. Oguz: A fully polynomial approximation algorithm for the 0-1 knapsack problem. *European Journal of Operational Research*, **8** (1981), 270–273.
- [14] S. Martello and P. Toth: *Knapsack Problems: Algorithms and Computer Implementations* (John Wiley and Sons, New York, 1990).
- [15] V.V. Vazirani: *Approximation Algorithms* (Springer, Berlin, 2001).
- [16] D.P. Williamson and D.B. Shmoys: *The Design of Approximation Algorithms* (Cambridge University Press, Cambridge, 2011).
- [17] T. Yamada and M. Futakawa: Heuristic and reduction algorithms for the knapsack sharing problem. *Computers & Operations Research*, **24** (1997), 961–967.
- [18] T. Yamada, M. Futakawa, and S. Kataoka: Some exact algorithms for the knapsack sharing problem. *European Journal of Operational Research*, **106** (1998), 177–183.

Seiji Kataoka  
Department of Computer Science  
National Defense Academy  
1-10-20 Hashirimizu, Yokosuka,  
Kanagawa 239-8686, Japan  
E-mail: seiji@nda.ac.jp