

# Approximability of Two Variants of Multiple Knapsack Problems

Shuichi Miyazaki<sup>1</sup>, Naoyuki Morimoto<sup>2</sup>, and Yasuo Okabe<sup>1</sup>

<sup>1</sup> Academic Center for Computing and Media Studies, Kyoto University,

<sup>2</sup> Institute for Integrated Cell-Material Sciences (iCeMS), Kyoto University,  
Yoshida-honmachi, Sakyo-ku, Kyoto 606-8501 Japan

shuichi@media.kyoto-u.ac.jp

nmorimoto@icems.kyoto-u.ac.jp

okabe@i.kyoto-u.ac.jp

**Abstract.** This paper considers two variants of Multiple Knapsack Problems. The first one is the Multiple Knapsack Problem with Assignment Restrictions and Capacity Constraints (MK-AR-CC). In the MK-AR-CC( $k$ ) (where  $k$  is a positive integer), a subset of knapsacks is associated with each item and the item can be packed into only those knapsacks (Assignment Restrictions). Furthermore, the size of each knapsack is at least  $k$  times the largest item assignable to the knapsack (Capacity Constraints). The MK-AR-CC( $k$ ) is NP-hard for any constant  $k$ . In this paper, we give a polynomial-time  $(1 + \frac{2}{k+1} + \epsilon)$ -approximation algorithm for the MK-AR-CC( $k$ ), and give a lower bound on the approximation ratio of our algorithm by showing an integrality gap of  $(1 + \frac{1}{k} - \epsilon)$  for the IP formulation we use in our algorithm, where  $\epsilon$  is an arbitrary small positive constant. The second problem is the Splittable Multiple Knapsack Problem with Assignment Restrictions (S-MK-AR), in which the size of items may exceed the capacity of knapsacks and items can be split and packed into multiple knapsacks. We show that approximating the S-MK-AR with the ratio of  $n^{1-\epsilon}$  is NP-hard even when all the items have the same profit, where  $n$  is the number of items and  $\epsilon$  is an arbitrary positive constant.

**Keywords:** Multiple Knapsack Problem, Assignment Restrictions, Approximation Algorithms

## 1 Introduction

This paper considers two variants of Multiple Knapsack Problems, motivated by efficient power allocation in recent and future power networks. Efficient usage of natural power sources, e.g. solar power or wind power, has been studied actively with the spread of in-home power generations such as solar panels. These kinds of relatively small sources are generally called distributed power sources. Future power networks are supposed to include various distributed generations in addition to conventional commercial power sources, e.g. fossil fuel plants or

nuclear power plants, and it is strongly desired these power sources are utilized effectively. Power sources have various characteristics such as cost, stability, and CO<sub>2</sub> emission. For example, power from fossil fuel plants is stable but relatively costly, while power from in-home solar panels is low-cost but unstable because it depends on weather conditions. Power consuming devices also have characteristics on quality of power that they require. For example, a desktop PC needs stable power, therefore it should be supplied from commercial sources, whereas a laptop with a battery accepts power from solar panels, because it can work with the battery even when the solar panels fails to generate stable power due to weather conditions. Therefore, it is desired to match power sources and power consuming devices in an appropriate manner [13, 15].

Power allocation can be naturally formalized as a combinatorial optimization problem as follows: There are power devices and power sources. Each power device  $d$  has two values, the profit  $p(d)$  and the power consumption  $c(d)$ , meaning that using the device  $d$  requires the power of  $c(d)$ , and if  $d$  can be used we gain the profit of  $p(d)$ . Each power source  $s$  has a capacity, that is a maximum power  $s$  can supply. Our goal is to allocate devices to sources so as to maximize the sum of the profits of allocated devices, while keeping the capacity constraint of each source. This problem can be viewed as the *Multiple Knapsack Problem (MK)* by regarding power sources as knapsacks and power consuming devices as items. Note that in the above mentioned characteristics-based power allocation, a device can be allocated to only a power source whose power quality matches the requirement of the device. One of the suitable extensions of the MK in this scenario is the *Multiple Knapsack Problem with Assignment Restriction (MK-AR)*, where a subset of knapsacks is associated with each item and the item can be packed into only those knapsacks. The MK-AR is NP-hard since it is a generalization of the classical Knapsack Problem. As for the approximability, Nutov et al. [12] showed a simple 2-approximation algorithm for the MK-AR. They also proposed an  $\frac{e}{e-1}$ -approximation algorithm for the fixed-profit Generalized Assignment Problem (GAP), which includes the MK-AR as a special case.

**Our Results.** In this paper, we consider two extensions of the MK-AR based on the observations on real power networks.

First, the capacity of power sources, such as a commercial power source, is usually much larger than the power consumption of devices. Therefore, it is reasonable to consider instances in which item sizes are small and capacities of knapsacks are large. The problem we propose in this context is the *Multiple Knapsack Problem with Assignment Restrictions and Capacity Constraints (MK-AR-CC)*. In the MK-AR-CC( $k$ ) (where  $k$  is a positive integer), the size of each knapsack is at least  $k$  times the largest item assignable to the knapsack. It is easy to see that the MK-AR-CC( $k$ ) is NP-hard for any constant  $k$  (by a straightforward reduction from the classical Knapsack Problem). In this paper we extend Nutov et al.'s 2-approximation algorithm [12] and give a polynomial-time  $\left(1 + \frac{2}{k+1} + \epsilon\right)$ -approximation algorithm for the MK-AR-CC( $k$ ). (In the case of  $k = 1$ , our algorithm is equivalent to Nutov et al.'s algorithm and hence the approximation ratio is 2 rather than  $2 + \epsilon$ .) We also give a lower bound on the

approximation ratio of our algorithm by showing an integrality gap of  $(1 + \frac{1}{k} - \epsilon)$  for the IP formulation we use in our algorithm, where  $\epsilon$  is an arbitrary positive constant.

The second scenario is that the capacity of distributed power sources is relatively small and one device may need to be allocated to two or more sources. We therefore consider the *Splittable Multiple Knapsack Problem with Assignment Restrictions (S-MK-AR)*, in which the size of items may exceed the capacity of knapsacks and one item can be split and packed into multiple knapsacks. We show that approximating the S-MK-AR with the ratio of  $n^{1-\epsilon}$  is NP-hard even when all the items have the same profit, where  $n$  is the number of items and  $\epsilon$  is an arbitrary positive constant.

**Related Work.** The MK-AR is a special case of the Generalized Assignment Problems (GAP). Approximation algorithms for the GAP and their variants have been studied actively. Shmoys and Tardos [14] presented a 2-approximation algorithm for the GAP. Later, Fleischer et al. [6] derived an  $\frac{e}{e-1}$ -approximation algorithm for restricted instances of Separable Assignment Problems that includes the GAP as a special case. Feige and Vondrak [5] have broken this barrier using randomization; they presented a randomized  $(\frac{e}{e-1} - \epsilon)$ -approximation algorithm for the GAP for some absolute constant  $\epsilon > 0$ . Cohen et al. [3] showed a combinatorial translation of any algorithm for the single knapsack problem into an approximation algorithm for the GAP, and showed a  $(1 + \alpha)$ -approximation algorithm for the GAP, where  $\alpha$  is an approximation ratio for the single knapsack problem.

Dawande et al. [4] showed a combinatorial 2-approximation algorithm for the restricted case of the MK-AR where the size of an item is equal to its profit. Approximation algorithms for other restricted instances of the MK-AR have been studied as well [1, 2].

## 2 The Multiple Knapsack Problem with Assignment Restrictions and Capacity Constraints

### 2.1 Problem Formulation

We define the Multiple Knapsack Problem with Assignment Restrictions (MK-AR) as follows. Its input is a bipartite graph  $G = (I, J, E)$  with a set of edges  $E$  between  $I$  and  $J$ . Vertices of  $I = \{a_1, a_2, \dots, a_n\}$  correspond to items, and vertices of  $J = \{b_1, b_2, \dots, b_m\}$  correspond to knapsacks. Item  $a \in I$  is assignable to knapsack  $b \in J$  only if  $(a, b) \in E$ . For each item  $a \in I$ , the *profit* and the *size* of  $a$ , denoted by  $p(a)$  and  $\ell(a)$  respectively, are associated. For each knapsack  $b \in J$ , its *capacity*  $c(b)$  is associated. A feasible solution of MK-AR is an assignment of items to knapsacks such that, for each  $b$ , the total size of assigned items to knapsack  $b$  is at most  $c(b)$ . The goal of the MK-AR is to maximize the total profit of assigned items. The MK-AR-CC( $k$ ) is a restriction of the MK-AR in which any instance satisfies the capacity constraints, namely,  $c(b) \geq k\ell(a)$  for any  $a$  and  $b$  such that  $(a, b) \in E$ .

For convenience, we define the profit and the size of an edge  $e$  as the profit and the size, respectively, of the item incident to  $e$ . Formally, the edge  $e = (a, b)$  has the profit  $p(e) = p(a)$  and the size  $\ell(e) = \ell(a)$ . For a vertex  $v$  of  $G$ ,  $\delta(v)$  denotes the set of edges that are incident to vertex  $v$ . Then the MK-AR can be formulated as an integer program (IP) as follows, where  $x_e$  is a decision variable;

$$\begin{aligned} & \max \sum_{e \in E} \frac{p(e)}{\ell(e)} x_e \\ \text{s.t. } & \sum_{e \in \delta(b)} x_e \leq c(b), \forall b \in J \\ & \sum_{e \in \delta(a)} x_e \leq \ell(a), \forall a \in I \\ & x_e \in \{0, \ell(e)\}, \forall e \in E \end{aligned}$$

The LP-relaxation of the MK-AR is defined by replacing the last constraint of IP formulation by “ $x_e \leq \ell(e), \forall e \in E$ ”.

## 2.2 Algorithm Match-and-FPTAS

The following corollary by Nutov et al. [12] is crucial for constructing and analyzing our algorithm. For a feasible solution  $x$  of the relaxation problem of the MK-AR, let  $F(x)$  be the graph that consists of the set of *fractional edges* in  $x$ , namely, the set of edges  $e$  such that  $0 < x_e < \ell(e)$ , and their endpoint vertices.

**Corollary 1.** (Nutov et al. [12]) *Given a feasible solution  $x$  to the LP relaxation of the MK-AR, we can find in  $O(|E(G)|^2)$  time a feasible solution  $z$  such that (i)  $\sum_{e \in E} \frac{p(e)}{\ell(e)} z_e \geq \sum_{e \in E} \frac{p(e)}{\ell(e)} x_e$ , (ii)  $F(z)$  is a forest, and (iii) in any connected component of  $F(z)$ , at most one leaf belongs to  $I$ .*

A formal description of our approximation algorithm **Match-and-FPTAS**, which is based on Nutov et al.’s algorithm for the MK-AR [12], is given in Algorithm 1. It first obtains an optimal solution  $x^*$  of an LP relaxation of the MK-AR-CC( $k$ ), using a polynomial time algorithm for linear programming [11, 7]. It then constructs a solution  $z$  from  $x^*$  using Corollary 1. In Step 3, for each knapsack  $b_j$ , we construct a single knapsack problem  $I_j$  consisting of knapsack  $b_j$ , *full items*, and at most one *matched item*. Full items are those assigned to  $b_j$  in  $x^*$  by an integral edge (i.e., an edge  $e$  such that  $x_e = \ell(e)$ ). A matched item is defined as follows. We construct a maximum cardinality matching  $M$  in  $F(z)$  (the graph that consists of the set of fractional edges in the solution  $z$  and their endpoint vertices) using the Hungarian method [9]. Then the matched item is the one matched with  $b_j$  in  $M$  if any. In Step 4, **Match-and-FPTAS** obtains a near-optimal solution for each single knapsack problem  $I_j$ , using an FPTAS [10, 8], and finally in Step 5, it outputs the union of selected items for each solution of  $I_j$ .

---

**Algorithm 1** Algorithm Match-and-FPTAS for the MK-AR-CC( $k$ )

---

**Step 1.** Obtain an optimal solution  $x^*$  of a relaxed instance of the MK-AR-CC( $k$ ), using a polynomial time algorithm for linear programming.

**Step 2.** Construct a solution  $z$  from  $x^*$  as shown in Corollary 1.

**Step 3.** Construct instances of the single knapsack problem by matching fractional items in  $F(z)$  to knapsacks.

**Step 4.** Using an FPTAS, obtain near-optimal solution for each instance of the single knapsack problem derived in Step 3.

**Step 5.** Output the union of the solutions obtained in Step 4 as a solution for the original MK-AR-CC( $k$ ) instance.

---

**2.3 Analysis of the Approximation Ratio of Match-and-FPTAS**

In our analysis, we focus on an instance  $I_j$  of the single knapsack problem derived in Step 3, and analyze its optimal solution.

For a set  $S$  of items, let  $\ell(S)$  and  $p(S)$  denote the total size and profit, respectively, of items in  $S$ . Let  $A_j$  denote the set of items in  $I_j$ , and let  $m_j$  denote, if any, the matched item of  $I_j$  derived in Step 3. Note that, if there does exist the matched item, then  $\ell(A_j) < c(b_j) + \ell_{max_j}$  holds where  $\ell_{max_j}$  is the maximum size of the items assignable to knapsack  $b_j$ , since the total size of the full items is less than  $c(b_j)$  and  $\ell(m_j) \leq \ell_{max_j}$  by definition. Let  $OPT_j$  be an optimal solution of  $I_j$ , and  $X_j$  be the set of items packed in the knapsack  $b_j$  in  $OPT_j$ . Define  $Y_j = A_j \setminus X_j$  as the set of items not packed in  $OPT_j$ .

First, we show some properties of an optimal solution  $OPT_j$ .

**Lemma 1.** For any  $S \subseteq X_j$  such that  $\ell(S) \geq \ell_{max_j}$ ,  $p(S) \geq p(Y_j)$ .

*Proof.* If we remove  $S$  from  $OPT_j$ , the vacancy of the knapsack is  $\ell(S) + (c(b_j) - \ell(X_j)) \geq \ell_{max_j} + (c(b_j) - \ell(X_j))$ . On the other hand,  $\ell(Y_j) = \ell(A_j) - \ell(X_j) = (\ell(A_j) - c(b_j)) + (c(b_j) - \ell(X_j)) < \ell_{max_j} + (c(b_j) - \ell(X_j))$ . Therefore, we can replace the item set  $S$  by  $Y_j$  in  $OPT_j$  to obtain another feasible solution. If  $p(S) < p(Y_j)$ , the profit of the new solution is larger than that of  $OPT_j$ , contradicting the optimality of  $OPT_j$ .  $\square$

**Lemma 2.** For any  $S \subseteq X_j$  and a positive integer  $k'$  such that  $\ell(S) \geq k' \ell_{max_j}$ ,  $p(S) \geq \left\lceil \frac{k'}{2} \right\rceil p(Y_j)$ .

*Proof.* Since the size of any item is at most  $\ell_{max_j}$ , we can partition  $S$  into at least  $\left\lceil \frac{k' \ell_{max_j}}{2 \ell_{max_j}} \right\rceil = \left\lceil \frac{k'}{2} \right\rceil$  subsets  $S_1, S_2, \dots, S_z$  ( $z \geq \left\lceil \frac{k'}{2} \right\rceil$ ), such that  $\ell_{max_j} \leq \ell(S_i) < 2 \ell_{max_j}$  for all  $i$ . Since each  $S_i$  satisfies the condition of Lemma 1, we have that  $p(S_i) \geq p(Y_j)$ . Hence,  $p(S) = \sum_{i=1}^z p(S_i) \geq \left\lceil \frac{k'}{2} \right\rceil p(Y_j)$ .  $\square$

**Lemma 3.** Suppose  $|Y_j| = 1$ . Then, for any  $S \subseteq X_j$  such that  $\ell(S) + (c(b_j) - \ell(X_j)) \geq \ell_{max_j}$ ,  $p(S) \geq p(Y_j)$ .

*Proof.* The proof goes like that of Lemma 1. If we remove  $S$  from  $OPT_j$ , the vacancy of the knapsack is  $\ell(S) + (c(b_j) - \ell(X_j)) \geq \ell_{max_j}$ . Since  $|Y_j| = 1$  and hence  $\ell(Y_j) \leq \ell_{max_j}$ , we can replace  $S$  by  $Y_j$  to obtain a feasible solution. If  $p(S) < p(Y_j)$ , the new solution is better than  $OPT_j$ , a contradiction.  $\square$

**Lemma 4.** *Suppose  $|Y_j| \geq 2$ . Then, for any  $S \subseteq X_j$  such that  $\ell(S) + (c(b_j) - \ell(X_j)) \geq \ell_{max_j}$ ,  $p(S) \geq \frac{1}{2}p(Y_j)$ .*

*Proof.* As before, if we remove all the items in  $S$  from  $OPT_j$ , we have the vacancy of  $\ell(S) + (c(b_j) - \ell(X_j)) \geq \ell_{max_j}$ . Let  $d$  denote an item with the lowest profit in  $Y_j$ . Then  $p(d) \leq \frac{p(Y_j)}{|Y_j|} \leq \frac{1}{2}p(Y_j)$  holds. Since  $X_j$  does not contain  $d$ ,  $\ell(X_j) + \ell(d) > c(b_j)$  (otherwise, we can add  $d$  to  $OPT_j$  to get a better solution). Since  $\ell(Y_j \setminus \{d\}) = \ell(A_j) - (\ell(X_j) + \ell(d)) < \ell(A_j) - c(b_j) < \ell_{max_j}$ , we can pack all the items in  $Y_j \setminus \{d\}$  if we remove  $S$  from  $OPT_j$ . Hence, by the optimality of  $OPT_j$ ,  $p(S) \geq p(Y_j) - p(d) \geq p(Y_j) - \frac{1}{2}p(Y_j) = \frac{1}{2}p(Y_j)$ .  $\square$

The following lemma is crucial to the analysis.

**Lemma 5.**  $\frac{p(A_j)}{p(X_j)} \leq 1 + \frac{2}{k+1}$ .

*Proof.* Since  $\frac{p(A_j)}{p(X_j)} = 1 + \frac{p(Y_j)}{p(X_j)}$ , it suffices to show that  $p(X_j) \geq \frac{k+1}{2}p(Y_j)$ . If  $|Y_j| = 0$ , then  $p(Y_j) = 0$  and the above inequality is satisfied trivially. Hence, from now on, we assume that  $|Y_j| \geq 1$ . Note that  $\ell(X_j) > c(b_j) - \ell_{max_j}$ , since otherwise, at least one item in  $Y_j$  could be packed in the knapsack  $b_j$ , contradicting the optimality of  $OPT_j$ .

We will do a case analysis depending on whether  $m_j \in X_j$  or not. Let  $k_1$  denote the positive integer such that  $k_1 \leq \frac{c(b_j)}{\ell_{max_j}} < k_1 + 1$ , i.e. the integer part of the ratio of  $c(b_j)$  to  $\ell_{max_j}$ . Note that  $k \leq k_1$  holds by the capacity constraint.

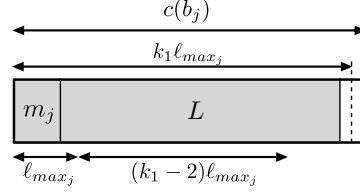
**Case 1.  $m_j \in X_j$ .** In this case, all the items in  $Y_j$  are full items. Note that all the full items in  $I_j$  can be packed in the knapsack  $b_j$ . Hence, if we remove  $m_j$  from  $OPT_j$  and add all the items in  $Y_j$ , the result is a feasible solution. Therefore, by the optimality of  $OPT_j$ ,  $p(m_j) \geq p(Y_j)$  holds. Hereafter, we will do a case analysis depending on  $k_1$ .

**Case 1-(i).  $k_1 = 1$ .** By the optimality of  $OPT_j$ ,  $p(X_j) \geq p(m_j) \geq p(Y_j) = \frac{k_1+1}{2}p(Y_j) \geq \frac{k+1}{2}p(Y_j)$ .

**Case 1-(ii).  $k_1 = 2$ .** Let  $L = X_j \setminus \{m_j\}$ . Since  $|Y_j| \geq 1$  and  $\ell(L) + (c(b_j) - \ell(X_j)) = c(b_j) - \ell(m_j) \geq k_1\ell_{max_j} - \ell_{max_j} = \ell_{max_j}$ ,  $p(L) \geq \frac{1}{2}p(Y_j)$  holds by Lemmas 3 and 4. Hence,  $p(X_j) = p(m_j) + p(L) \geq p(Y_j) + \frac{1}{2}p(Y_j) = \frac{3}{2}p(Y_j) = \frac{k_1+1}{2}p(Y_j) \geq \frac{k+1}{2}p(Y_j)$ .

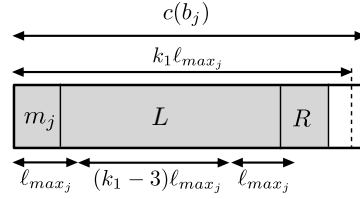
**Case 1-(iii).  $k_1$  is odd and  $k_1 \geq 3$ .** Let  $L = X_j \setminus \{m_j\}$  as shown in Fig. 1. Since  $\ell(m_j) \leq \ell_{max_j}$  and  $\ell(m_j) + \ell(L) = \ell(X_j) > c(b_j) - \ell_{max_j}$ ,  $\ell(L) > c(b_j) - \ell_{max_j} - \ell(m_j) \geq (k_1 - 2)\ell_{max_j}$ . Therefore, we obtain  $p(L) \geq \lceil \frac{k_1-2}{2} \rceil p(Y_j) = \frac{k_1-1}{2}p(Y_j)$  by Lemma 2. Hence,  $p(X_j) = p(m_j) + p(L) \geq p(Y_j) + \frac{k_1-1}{2}p(Y_j) \geq \frac{k_1+1}{2}p(Y_j) \geq \frac{k+1}{2}p(Y_j)$ .

**Case 1-(iv).  $k_1$  is even and  $k_1 \geq 4$ .** We partition  $X_j$  into three subsets  $\{m_j\}$ ,  $L$ , and  $R$  as follows:  $L$  is a subset of  $X_j \setminus \{m_j\}$  that satisfies  $(k_1 - 2)\ell_{max_j} \leq$



**Fig. 1.** Case 1-(iii):  $m_j \in X_j$  and  $k_1 (\geq 3)$  is odd.

$\ell(m_j) + \ell(L) < (k_1 - 1)\ell_{max_j}$  and  $R = X_j \setminus (L \cup \{m_j\})$  (see Fig. 2). Since  $\ell(m_j) \leq \ell_{max_j}$ , we have  $\ell(L) \geq (k_1 - 3)\ell_{max_j}$ . Therefore,  $p(L) \geq \lceil \frac{k_1 - 3}{2} \rceil p(Y_j) = (\frac{k_1}{2} - 1)p(Y_j)$  by Lemma 2. Also, since  $|Y_j| \geq 1$  and  $\ell(R) + (c(b_j) - \ell(X_j)) \geq \ell_{max_j}$ ,  $p(R) \geq \frac{1}{2}p(Y_j)$  holds by Lemmas 3 and 4. Hence,  $p(X_j) = p(m_j) + p(L) + p(R) \geq p(Y_j) + (\frac{k_1}{2} - 1)p(Y_j) + \frac{1}{2}p(Y_j) = \frac{k_1 + 1}{2}p(Y_j) \geq \frac{k_1 + 1}{2}p(Y_j)$ .



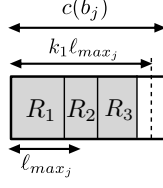
**Fig. 2.** Case 1-(iv):  $m_j \in X_j$  and  $k_1 (\geq 4)$  is even.

**Case 2.  $m_j \notin X_j$ .** In this case,  $Y_j = \{m_j\}$  because all the full items can be packed in the knapsack. Again, we will do a case analysis depending on  $k_1$ .

**Case 2-(i).  $k_1 = 1$ .** By the optimality of  $OPT_j$ ,  $p(X_j) \geq p(m_j) = p(Y_j) = \frac{k_1 + 1}{2}p(Y_j) \geq \frac{k_1 + 1}{2}p(Y_j)$ .

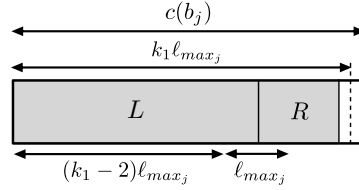
**Case 2-(ii).  $k_1 = 2$ .** We partition  $X_j$  into three subsets  $R_1$ ,  $R_2$ , and  $R_3$  as follows (see Fig. 3): First we sort all the items in  $X_j$  as  $\{a_1, a_2, \dots, a_{|X_j|}\}$  in descending order of sizes. Then we define  $R_1 = \{a_1, a_2, \dots, a_{q-1}\}$ ,  $R_2 = \{a_q\}$ , and  $R_3 = \{a_{q+1}, a_{q+2}, \dots, a_{|OPT'_j|}\}$ , where  $q$  is the positive integer that satisfies  $\ell(R_1) \leq \ell_{max_j} < \ell(R_1) + \ell(R_2)$ . Note that  $\ell(R_1) \geq \ell(R_2)$  holds.

Since  $\ell(R_1) + \ell(R_2) > \ell_{max_j}$ , we obtain  $p(R_1) + p(R_2) \geq p(Y_j)$  by Lemma 1. Since  $|Y_j| = 1$  and  $\ell(R_2) + \ell(R_3) + (c(b_j) - \ell(X_j)) \geq c(b_j) - \ell(R_1) \geq k_1 \ell_{max_j} - \ell_{max_j} = \ell_{max_j}$ , we obtain  $p(R_2) + p(R_3) \geq p(Y_j)$  by Lemma 3. Also, since  $|Y_j| = 1$  and  $\ell(R_3) + \ell(R_1) + (c(b_j) - \ell(X_j)) \geq \ell(R_3) + \ell(R_2) + (c(b_j) - \ell(X_j)) \geq \ell_{max_j}$ , we obtain  $p(R_3) + p(R_1) \geq p(Y_j)$  by Lemma 3. Hence,  $(p(R_1) + p(R_2)) + (p(R_2) + p(R_3)) + (p(R_3) + p(R_1)) \geq 3p(Y_j)$ , resulting that  $p(X_j) = p(R_1) + p(R_2) + p(R_3) \geq \frac{3}{2}p(Y_j) = \frac{k_1 + 1}{2}p(Y_j) \geq \frac{k_1 + 1}{2}p(Y_j)$ .



**Fig. 3.** Case 2-(ii):  $m_j \notin X_j$  and  $k_1 = 2$ .

**Case 2-(iii).  $k_1$  is odd and  $k_1 \geq 3$ .** We partition  $X_j$  into two subsets  $L$  and  $R$  as follows:  $L$  is a subset of  $X_j$  that satisfies  $(k_1 - 2)\ell_{max_j} \leq \ell(L) < (k_1 - 1)\ell_{max_j}$ , and  $R = X_j \setminus L$  (see Fig. 4). Since  $\ell(L) \geq (k_1 - 2)\ell_{max_j}$ ,  $p(L) \geq \lceil \frac{k_1 - 2}{2} \rceil p(Y_j) = \frac{k_1 - 1}{2} p(Y_j)$  holds by Lemma 2. Since  $|Y_j| = 1$  and  $\ell(R) + (c(b_j) - \ell(X_j)) = c(b_j) - \ell(L) > k_1 \ell_{max_j} - (k_1 - 1)\ell_{max_j} = \ell_{max_j}$ , we have that  $p(R) \geq p(Y_j)$  by Lemma 3. Hence,  $p(X_j) = p(L) + p(R) \geq \frac{k_1 - 1}{2} p(Y_j) + p(Y_j) = \frac{k_1 + 1}{2} p(Y_j) \geq \frac{k_1 + 1}{2} p(Y_j)$ .

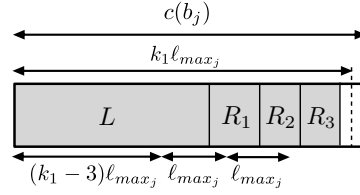


**Fig. 4.** Case 2-(iii):  $m_j \notin X_j$  and  $k_1 (\geq 3)$  is odd.

**Case 2-(iv).  $k_1$  is even and  $k_1 \geq 4$ .** We partition  $X_j$  into four subsets  $L$ ,  $R_1$ ,  $R_2$ , and  $R_3$  as follows (see also Fig. 5): First we sort all the items in  $X_j$  as  $\{a_1, a_2, \dots, a_{|X_j|}\}$  in descending order of sizes. Then we partition them as  $L = \{a_1, a_2, \dots, a_{q-1}\}$ ,  $R_1 = \{a_q, a_{q+1}, \dots, a_{r-1}\}$ ,  $R_2 = \{a_r\}$ , and  $R_3 = \{a_{r+1}, a_{r+2}, \dots, a_{|X_j|}\}$ , where  $q$  and  $r$  are positive integers that satisfy  $(k_1 - 3)\ell_{max_j} \leq \ell(L) < (k_1 - 2)\ell_{max_j}$ ,  $(k_1 - 2)\ell_{max_j} \leq \ell(L) + \ell(R_1) < (k_1 - 1)\ell_{max_j}$ , and  $(k_1 - 1)\ell_{max_j} \leq \ell(L) + \ell(R_1) + \ell(R_2)$ . Note that  $\ell(R_1) \geq \ell(R_2)$  holds.

Since  $\ell(L) \geq (k_1 - 3)\ell_{max_j}$ , we obtain  $p(L) \geq \lceil \frac{k_1 - 3}{2} \rceil p(Y_j) = (\frac{k_1}{2} - 1)p(Y_j)$  by Lemma 2. Since  $\ell(R_1) + \ell(R_2) = (\ell(L) + \ell(R_1) + \ell(R_2)) - \ell(L) > (k_1 - 1)\ell_{max_j} - (k_1 - 2)\ell_{max_j} = \ell_{max_j}$ , we obtain  $p(R_1) + p(R_2) \geq p(Y_j)$  by Lemma 1. Since  $|Y_j| = 1$  and  $\ell(R_2) + \ell(R_3) + (c(b_j) - \ell(X_j)) \geq c(b_j) - (\ell(L) + \ell(R_1)) > k_1 \ell_{max_j} - (k_1 - 1)\ell_{max_j} = \ell_{max_j}$ , we have  $p(R_2) + p(R_3) \geq p(Y_j)$  by Lemma 3. Also, since  $|Y_j| = 1$  and  $\ell(R_3) + \ell(R_1) + (c(b_j) - \ell(X_j)) \geq \ell(R_3) + \ell(R_2) + (c(b_j) - \ell(X_j)) > \ell_{max_j}$ , we have that  $p(R_3) + p(R_1) \geq p(Y_j)$  by Lemma 3. Therefore, it follows that  $(p(R_1) + p(R_2)) + (p(R_2) + p(R_3)) + (p(R_3) + p(R_1)) \geq 3p(Y_j)$ , resulting that





**Fig. 5.** Case 2-(iv):  $m_j \notin X_j$  and  $k_1 (\geq 4)$  is even.

$p(R_1) + p(R_2) + p(R_3) \geq \frac{3}{2}p(Y_j)$ . Hence,  $p(X_j) = p(L) + p(R_1) + p(R_2) + p(R_3) \geq (\frac{k_1}{2} - 1)p(Y_j) + \frac{3}{2}p(Y_j) = \frac{k_1 + 1}{2}p(Y_j) \geq \frac{k + 1}{2}p(Y_j)$ .  $\square$

**Theorem 1.** *Match-and-FPTAS is a  $(1 + \frac{2}{k+1} + \epsilon)$ -approximation algorithm for the MK-AR-CC( $k$ ), where  $\epsilon$  is an arbitrary positive constant.*

*Proof.* Let  $p(OPT)$  and  $p(LPOPT)$  denote the profits of optimal solutions for the MK-AR-CC( $k$ ) and its LP-relaxation, respectively. Note that  $p(LPOPT) \leq \sum_j p(A_j)$ . Also, let  $p(MF)$  denote the profit of the solution obtained by Match-and-FPTAS, and  $p(MF_j)$  denote the profit of the solution for  $I_j$  obtained in Step 4 of Match-and-FPTAS. Let  $\epsilon'$  be a positive constant that satisfies  $\epsilon' \leq \frac{\epsilon(k+1)}{k+3+\epsilon(k+1)}$ . Since we use an FPTAS for the knapsack problem in Step 4, we can have  $p(MF_j) \geq (1 - \epsilon')p(X_j)$ . By Lemma 5, we obtain

$$\begin{aligned} \frac{p(OPT)}{p(MF)} &\leq \frac{p(LPOPT)}{p(MF)} \leq \frac{\sum_j p(A_j)}{\sum_j p(MF_j)} \\ &\leq \max_j \left\{ \frac{p(A_j)}{p(MF_j)} \right\} \leq \max_j \left\{ \frac{p(A_j)}{(1 - \epsilon')p(X_j)} \right\} \\ &\leq \frac{1}{1 - \epsilon'} \left( 1 + \frac{2}{k+1} \right) \\ &\leq 1 + \frac{2}{k+1} + \epsilon. \end{aligned}$$

$\square$

## 2.4 Integrality Gap of the IP Formulation Used in Match-and-FPTAS

**Theorem 2.** *The integrality gap of the IP formulation of the MK-AR-CC( $k$ ) used in Match-and-FPTAS is at least  $1 + \frac{1}{k} - \epsilon$  for any positive constant  $\epsilon$ .*

*Proof.* We consider an instance of the MK-AR-CC( $k$ ) including  $k + 1$  items with profit 1 and size 1, and one knapsack with capacity  $k + 1 - \epsilon$ . The profit of an optimal solution for LP-relaxation is  $k + 1 - \epsilon$ , while that for the original MK-AR-CC( $k$ ) is  $k$ . Hence, the integrality gap is  $1 + \frac{1}{k} - \frac{\epsilon}{k} \geq 1 + \frac{1}{k} - \epsilon$ .  $\square$

### 3 Splittable Multiple Knapsack Problem with Assignment Restrictions

#### 3.1 Problem Definition

We define the Splittable Multiple Knapsack Problem with Assignment Restrictions (S-MK-AR) as follows. We are given a bipartite graph  $G = (I, J, E)$ , where  $I$  and  $J$  correspond to items and knapsacks, respectively. A vertex (item)  $a \in I$  has the *size*  $\ell(a)$  and the *profit*  $p(a)$ . A vertex (knapsack)  $b \in J$  has the *capacity*  $c(b)$ . A feasible solution of S-MK-AR is a non-negative weight on each  $e \in E$  such that the total weight of edges incident to a vertex  $b \in J$  is at most  $c(b)$  and the total weight of edges incident to a vertex  $a \in I$  is at most  $\ell(a)$ . If the total weight of incident edges of  $a$  equals  $\ell(a)$ , we say that the item  $a$  is *satisfied*. The *profit of a feasible solution* is the sum of the profits of the satisfied items, and the goal of the S-MK-AR is to maximize it.

#### 3.2 Approximation Hardness of the S-MK-AR

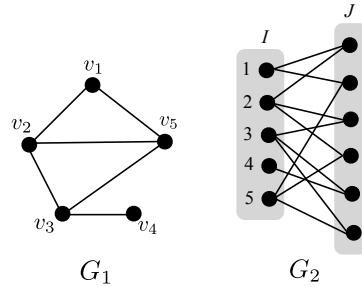
**Theorem 3.** *For any positive constant  $\epsilon$ , there is no polynomial-time  $n^{1-\epsilon}$ -approximating algorithm for the S-MK-AR unless  $P=NP$ , even if all the items have the same profit, where  $n$  is the number of items in an input.*

*Proof.* The proof is done by a reduction from the Maximum Independent Set Problem (MIS). For a graph  $G = (V, E)$ , a subset  $S \subseteq V$  is called an *independent set* if there is no edge between any pair of vertices in  $S$ . MIS is the problem of, given a graph  $G$ , finding an independent set of  $G$  with the maximum cardinality.

Given an instance  $G_1 = (V, E)$  of MIS, we construct an instance  $L_2$  of the S-MK-AR (whose underlying graph is  $G_2 = (I, J, E')$ ) as follows (an example of the reduction is illustrated in Fig. 6).

Without loss of generality, we can assume that  $G_1$  has no isolated vertex. Suppose that  $G_1$  has  $n$  vertices  $v_1, \dots, v_n$ , and  $m$  edges  $e_1, \dots, e_m$ . Then,  $I$  has  $n$  vertices  $a_1, \dots, a_n$ , where  $a_i$  corresponds to the vertex  $v_i$  of  $G_1$ . Similarly,  $J$  has  $m$  vertices  $b_1, \dots, b_m$ , where  $b_j$  corresponds to the edge  $e_j$  of  $G_1$ . Finally, we define  $E'$  as  $(a_i, b_j) \in E'$  if and only if the vertex  $v_i$  is incident to the edge  $e_j$  in  $G_1$ . The capacity of the knapsack  $b_j$  is  $c(b_j) = 1$  for every  $j$ . The profit and the size of an item  $a_i$  is  $p(a_i) = 1$  and  $\ell(a_i) = \deg(v_i)$ , respectively, where  $\deg(v_i)$  denotes the degree of  $v_i$ . Clearly this reduction can be done in polynomial time.

Suppose that  $S$  is an independent set of  $G_1$ . We will construct a solution  $T$  of  $L_2$  in such a way that if  $v_i \in S$  then all the edges in  $\delta(a_i)$  have weight 1, and if  $v_i \notin S$  then all the edges in  $\delta(a_i)$  have weight 0 (recall that  $\delta(a_i)$  is the set of edges incident to  $a_i$ ). Note that this is a feasible solution because, for any  $b_j \in J$ , at most one edge in  $\delta(b_j)$  has weight 1 (and other edge(s) have weight 0), since  $S$  is an independent set. Also, it is not hard to see that for any  $i$  such that  $v_i \in S$ , the item  $a_i$  is satisfied since all of its incident edges have weight 1. Therefore, the profit of  $T$  is  $|S|$ . This implies that  $OPT(L_2) \geq OPT(G_1)$ , where  $OPT(G_1)$  denotes the size of a maximum independent set of  $G_1$  and  $OPT(L_2)$  denotes the profit of an optimal solution of  $L_2$ .



**Fig. 6.** An example of the reduction from the Maximum Independent Set Problem to the S-MK-AR.

Next, consider a feasible solution  $T$  of  $L_2$ , and denote its profit  $p(T)$ . It is not hard to see from the above argument that if we choose a vertex  $v_i$  for every  $i$  such that  $a_i$  is satisfied by  $T$ , we have an independent set of  $G_1$  whose size is the same as  $p(T)$ . Therefore, from a feasible solution  $T$  of  $L_2$ , we can construct in polynomial time an independent set  $S$  such that  $|S| = p(T)$ .

Now, suppose that there is a polynomial time  $n^{1-\epsilon}$ -approximation algorithm  $ALG_1$  for the S-MK-AR. Consider the following approximation algorithm  $ALG_2$  for MIS: Given an instance  $G_1$  of MIS,  $ALG_2$  first transforms it to  $L_2$  using the above reduction. It then solves  $L_2$  using  $ALG_1$  and obtains a solution  $T$ . Finally,  $ALG_2$  transforms  $T$  into an independent set  $S$  of  $G_1$ . By the above arguments, we have that  $\frac{OPT(G_1)}{|S|} = \frac{OPT(L_2)}{p(T)} \leq n^{1-\epsilon}$ , that is,  $ALG_2$  is an  $n^{1-\epsilon}$ -approximation algorithm for MIS. It is known that existence of an  $n^{1-\epsilon}$ -approximation algorithm for MIS implies P=NP [16]. This completes the proof.  $\square$

## 4 Conclusion

In this paper, we have considered two variants of the Multiple Knapsack Problems, motivated by efficient power allocation in power networks. As future work, it would be challenging to fill the gap between upper and lower bounds on approximation ratios. It would also be interesting to formulate the problem in an online manner, considering the scenario where power consuming devices and power sources appear and disappear dynamically.

## Acknowledgements

The authors would like to thank anonymous reviewers for their helpful comments. This work was supported by JSPS KAKENHI Grant Number 24500013, JST Super Cluster Program, and NICT Advanced Telecommunication Research Fund.

## References

1. Aerts, J., Korst, J., Spieksma, F.: Approximation of a retrieval problem for parallel disks. In: Proceedings of the 5th Italian Conference on Algorithms and Complexity (CIAC). pp. 178–188 (2003)
2. Aerts, J., Korst, J., Spieksma, F., Verhaegh, W., Woeginger, G.: Random redundant storage in disk arrays: complexity of retrieval problems. *IEEE Transactions on Computers* 52(9), 1210–1214 (2003)
3. Cohen, R., Katzir, L., Raz, D.: An efficient approximation for the generalized assignment problem. *Information Processing Letters* 100(4), 162–166 (2006)
4. Dawande, M., Kalagnanam, J., Keskinocak, P., Salman, F.S., Ravi, R.: Approximation algorithms for the multiple knapsack problem with assignment restrictions. *Journal of Combinatorial Optimization* 4(2), 171–186 (2000)
5. Feige, U., Vondrak, J.: Approximation algorithms for allocation problems: Improving the factor of  $1 - 1/e$ . In: Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS). pp. 667–676 (2006)
6. Fleischer, L., Goemans, M.X., Mirrokni, V.S., Sviridenko, M.: Tight approximation algorithms for maximum separable assignment problems. *Mathematics of Operations Research* 36(3), 416–431 (2011)
7. Karmarkar, N.: A new polynomial-time algorithm for linear programming. In: Proceedings of the 16th Annual ACM Symposium on Theory of Computing (STOC). pp. 302–311 (1984)
8. Kellerer, H., Pferschy, U.: A new fully polynomial time approximation scheme for the knapsack problem. *Journal of Combinatorial Optimization* 3(1), 59–71 (1999)
9. Kuhn, H.W.: The hungarian method for the assignment problem. *Naval Research Logistics Quarterly* 2(1-2), 83–97 (1955)
10. Magazine, M.J., Oguz, O.: A fully polynomial approximation algorithm for the 0–1 knapsack problem. *European Journal of Operational Research* 8(3), 270–273 (1981)
11. Murty, K.G.: *Network Programming*. Prentice-Hall, Inc. (1992)
12. Nutov, Z., Beniaminy, I., Yuster, R.: A  $(1 - 1/e)$ -approximation algorithm for the generalized assignment problem. *Operations Research Letters* 34(3), 283–288 (2006)
13. Sakai, K., Okabe, Y.: Quality-aware energy routing toward on-demand home energy networking. In: Proceeding of the 2011 IEEE Consumer Communications and Networking Conference (CCNC). pp. 1041–1044 (2011)
14. Shmoys, D.B., Tardos, É.: An approximation algorithm for the generalized assignment problem. *Mathematical Programming* 62(1-3), 461–474 (1993)
15. Takuno, T., Kitamori, Y., Takahashi, R., Hikihara, T.: AC power routing system in home based on demand and supply utilizing distributed power sources. *Energies* 4(5), 717–726 (2011)
16. Zuckerman, D.: Linear degree extractors and the inapproximability of max clique and chromatic number. In: Proceedings of the 38th Annual ACM Symposium on Theory of Computing (STOC). pp. 681–690 (2006)