A Constant Factor Approximation for the Generalized Assignment Problem with Minimum Quantities and Unit Size Items

Marco Bender[1,⋆], Clemens Thielen[2], and Stephan Westphal[1]

[1] Institute for Numerical and Applied Mathematics, University of Goettingen, Lotzestr. 16-18, D-37083 Goettingen, Germany
{m.bender,s.westphal}@math.uni-goettingen.de
[2] Department of Mathematics, University of Kaiserslautern, Paul-Ehrlich-Str. 14, D-67663 Kaiserslautern, Germany
thielen@mathematik.uni-kl.de

Abstract. We consider a variant of the generalized assignment problem (GAP) where the items have unit size and the amount of space used in each bin is restricted to be either zero (if the bin is not opened) or above a given lower bound (a *minimum quantity*). This problem is known to be strongly NP-complete and does not admit a polynomial time approximation scheme (PTAS).

By using randomized rounding, we obtain a randomized 3.93-approximation algorithm, thereby providing the first nontrivial approximation result for this problem.

Keywords: generalized assignment problem, combinatorial optimization, approximation algorithms, randomized rounding.

1 Introduction

The generalized assignment problem (GAP) is a classical generalization of both the (multiple) knapsack problem and the bin packing problem. In the classical version of GAP (cf., for example, [1, 2]), one is given m bins, a capacity B_j for each bin j, and n items such that each item i has size $s_{i,j}$ and yields profit $p_{i,j}$ when packed into bin j. The goal is to find a feasible packing of the items into the bins that maximizes the total profit. The problem has many practical applications, for which we refer to [2] and the references therein.

Recently, Krumke and Thielen [3] introduced the generalized assignment problem with minimum quantities (GAP-MQ), which is a variation of the generalized assignment problem where the amount of space used in each bin is restricted to be either zero (if the bin is not opened) or above a given lower bound (a *minimum quantity*). This additional restriction is motivated from many practical

⋆ Partially supported by DFG RTG 1703 "Resource Efficiency in Interorganizational Networks".

K. Chatterjee and J. Sgall (Eds.): MFCS 2013, LNCS 8087, pp. 135–145, 2013.

packing problems where it does often not make sense to open an additional container (bin) if not at least a certain amount of space in it will be used. While it is not hard to see that it is NP-hard to compute any feasible solution with positive profit for the general version of GAP-MQ (and, hence, no polynomial time approximation algorithm exists for the problem unless P = NP), computing nontrivial feasible solutions is easy when all items have unit size. Due to its application in assigning students (unit size items) to seminars (bins) at a university such that the total satisfaction (profit) of the students is maximized, this special case of GAP-MQ where all items have unit size was termed *seminar assignment problem* (SAP) in [3] and is formally defined as follows:

Definition 1 (Seminar Assignment Problem (SAP))
INSTANCE: *The number n of items, m bins with capacities $B_1, \ldots, B_m \in \mathbb{N}$ and minimum quantities $q_1, \ldots, q_m \in \mathbb{N}$ (where $q_j \leq B_j \leq n$ for all $j = 1, \ldots, m$), and a profit $p_{i,j} \in \mathbb{N}$ resulting from assigning item i to bin j for $i = 1, \ldots, n$ and $j = 1, \ldots, m$.*
TASK: *Find an assignment of a subset of the items to the bins such that the number of items in each bin j is either zero (if bin j is not opened) or at least q_j and at most B_j and the total profit is maximized.*

Note that, in the above definition and throughout the paper, we always assume $\mathbb{N}$ to contain zero and denote the positive integers by $\mathbb{N}_+$.

Even though computing nontrivial feasible solutions for SAP is easy, a gap-preserving reduction from the 3-bounded 3-dimensional matching problem (3DM-3) given in [3] shows the existence of a constant $\epsilon_0 > 0$ such that it is strongly NP-hard to approximate SAP within a factor smaller than $(1 + \epsilon_0)$ even if all profits $p_{i,j}$ are in $\{0, 1\}$ and the minimum quantities and bin capacities of all bins are fixed to three. In particular, the problem does not admit a polynomial time approximation scheme (PTAS). Apart from these negative results, however, the approximability of SAP (and, in particular, the existence of a constant factor approximation) remained open. As most standard techniques for designing deterministic approximation algorithms fail for this problem due to the minimum quantity restrictions (cf. [3]), it natural to consider randomization and to ask whether a constant approximation ratio can be obtained by a randomized algorithm.

In this paper, we answer this question by presenting a randomized 3.93-approximation algorithm for SAP, which is the first nontrivial approximation result for this problem. Our randomized rounding algorithm uses a packing-based integer programming formulation, for which we show that the linear relaxation can be solved in polynomial time by using column generation. In particular, by using the probabilistic method (cf., for example, [4]), our result implies that the integrality gap of this formulation is no larger than 3.93.

1.1 Previous Work

The classical GAP is well-studied in literature. A comprehensive introduction to the problem can be found in [1]. A survey of algorithms for GAP is given in [2].

For a survey on different variants of assignment problems studied in literature, we refer to [5].

GAP is known to be APX-hard [6], but there exists a 2-approximation algorithm [7, 6]. Cohen et al. [8] showed how any polynomial time α-approximation algorithm for the knapsack problem can be translated into a polynomial time $(1+\alpha)$-approximation algorithm for GAP. A $(1, 2)$-approximation algorithm for the equivalent minimization version of GAP, in which assigning item i to bin j causes a cost $c_{i,j}$, was provided by Shmoys and Tardos [7]: For every feasible instance of GAP, their algorithm computes a solution that violates the bin capacities by at most a factor of 2 and whose cost is at most as large as the cost of the best solution that satisfies the bin capacities strictly.

GAP is a generalization of both the (multiple) knapsack problem (cf. [1, 6, 9]) and the bin packing problem (cf. [10–12]). The multiple knapsack problem is the special case of GAP where the size and profit of an item are independent of the bin (knapsack) it is packed into. The bin packing problem can be seen as the special case of the decision version of GAP in which all bins have the same capacity and all profits are one. The question of deciding whether a packing of total profit equal to the number of items exists is then equivalent to asking whether all items can be packed into the given number of bins.

A dual version of bin packing (often called *bin covering*) in which minimum quantities are involved was introduced in [13, 14]. Here, the problem is to pack a given set of items with sizes that do not depend on the bins so as to maximize the number of bins used, subject to the constraint that each bin contains items of total size at least a given threshold T (upper bin capacities are not considered due to the nature of the objective function). Hence, the bin covering problem can be seen as a variant of GAP-MQ in which the minimum quantity is the same for each bin and the objective is to maximize the number of bins used. Since any approximation algorithm with approximation ratio strictly smaller than 2 would have to solve the NP-complete partition problem when applied to instances in which the sizes of the items sum up to two, it follows that (unless $\mathsf{P} = \mathsf{NP}$) no polynomial time $(2 - \epsilon)$-approximation for bin covering exists for any $\epsilon > 0$. In contrast, the main result of Assmann et al. [14] is an $\mathcal{O}(n \log^2 n)$ time algorithm that yields an *asymptotic* approximation ratio of 4/3 for bin covering, while easier algorithms based on next fit and first fit decreasing are shown to yield asymptotic approximation ratios of 2 and 3/2, respectively. Later, an asymptotic PTAS [15] and an asymptotic FPTAS [16] for bin covering were developed.

Minimum quantities have recently been studied for minimum cost network flow problems [17–19]. In this setting, minimum quantities for the flow on each arc are considered, which results in the minimum cost flow problem becoming strongly NP-complete [18]. Moreover, it was shown in [18] that (unless $\mathsf{P} = \mathsf{NP}$) no polynomial time $g(|I|)$-approximation for the problem exists for any polynomially computable function $g : \mathbb{N}_+ \to \mathbb{N}_+$, where $|I|$ denotes the encoding length of the given instance. The special case of the maximum flow problem with minimum quantities has recently been studied in [20], where it was shown that the problem is strongly NP-hard to approximate in general, but admits a $(2 - \frac{1}{\lambda})$-approximation in the case of an identical minimum quantity λ on all arcs.

The generalized assignment problem with minimum quantities (GAP-MQ) and the seminar assignment problem (SAP) were introduced in [3], where it was shown that the general version of GAP-MQ does not admit any polynomial time approximation algorithm unless $\mathsf{P} = \mathsf{NP}$. For SAP, it was shown by a gap-preserving reduction from the 3-bounded 3-dimensional matching problem (3DM-3) that there exists a constant $\epsilon_0 > 0$ such that it is strongly NP-hard to approximate SAP within a factor smaller than $(1 + \epsilon_0)$ even if all profits $p_{i,j}$ are in $\{0, 1\}$ and the minimum quantities and bin capacities of all bins are fixed to three. In particular, the problem does not admit a polynomial time approximation scheme (PTAS). Apart from these negative results, however, the approximability of SAP (and, in particular, the existence of a constant factor approximation) remained open.

2 Overview of the Algorithm

Before we present our randomized rounding algorithm for SAP in detail, we give a brief overview of the different steps of our procedure and its analysis.

The algorithm is based on an integer programming formulation of SAP that is introduced in Section 3. For each bin j, the integer program contains a binary variable x_t for every feasible packing of j (i.e., for every assignment of $q_j \leq l \leq B_j$ items to bin j), where $x_t = 1$ means that packing t is selected for bin j. As we show in Theorem 1, the linear relaxation of this integer program can be solved in polynomial time by column generation even though it contains an exponential number of variables x_t.

After solving the linear relaxation of the integer program, our algorithm independently selects a packing for each bin by using the value of variable x_t in the optimal solution (scaled by a suitably chosen factor $\alpha \in [0, 1]$) as the probability of using packing t for the corresponding bin j. The expected profit of the set of packings obtained in this way is exactly α times the objective value of the optimal solution of the linear relaxation used for the rounding, but the set of packings will, in general, not correspond to a feasible integral solution as items may be packed several times into different bins. Hence, in order to obtain a feasible integral solution, we apply a clean-up procedure that works in two steps: In the first step, we discard a subset of the bins opened in order to ensure that the total number of places used in the bins is at most n. In the second step, we can then replace all remaining multiply assigned items in the solution by unassigned items in order to obtain a feasible integral solution. Overall, we show that, in expectation, the profit decreases by at most a factor 3.93 during the clean-up procedure, which yields the desired approximation guarantee.

3 An Integer Programming Formulation

We start by introducing the IP-formulation on which our randomized rounding algorithm is based.

Definition 2. *A* (feasible) packing *of bin j is an incidence vector of a subset of the items with cardinality at least q_j and at most B_j, i.e., a vector $t = (t_1, \ldots, t_n) \in \{0,1\}^n$ such that $q_j \leq \sum_{i=1}^n t_i \leq B_j$. The* profit *of t is $p_t := \sum_{i=1}^n p_{ij} \cdot t_i$. The set of all feasible packings of bin j will be denoted by $T(j)$ and we write $T := \dot{\bigcup}_{j=1}^m T(j)$.*

Using this definition, we can formulate SAP as the following integer program:

$$\max \quad \sum_{j=1}^{m} \sum_{t \in T(j)} x_t p_t \tag{1a}$$

$$\text{s.t.} \quad \sum_{t \in T(j)} x_t \leq 1 \qquad \forall\, j \in \{1, \ldots, m\} \tag{1b}$$

$$\sum_{j=1}^{m} \sum_{t \in T(j)} x_t t_i \leq 1 \qquad \forall\, i \in \{1, \ldots, n\} \tag{1c}$$

$$x_t \in \{0,1\} \qquad \forall\, t \in T \tag{1d}$$

Here, variable x_t for $t \in T(j)$ is one if and only if packing t is selected for bin j. Constraint (1b) ensures that at most one packing is selected for each bin while constraint (1c) ensures that each item is packed into at most one bin.

We now show that, even though the number of variables in IP (1) exponential in the encoding length of the given instance of SAP, we can solve its linear relaxation in polynomial time by using column generation. To this end, it suffices to show that we can find a column (packing) of minimum reduced cost in polynomial time, i.e., solve the pricing problem in polynomial time (cf. [21, 22]). Denoting the dual variables corresponding to the constraints (1b) by y_j, $j = 1, \ldots, m$, and the dual variables corresponding to the constraints (1c) by z_i, $i = 1, \ldots, n$, the reduced cost of a packing $t \in T(j)$ of bin j is

$$\bar{c}_t = p_t - y_j - \sum_{i=1}^{n} t_i z_i = -y_j + \sum_{i=1}^{n} t_i (p_{ij} - z_i).$$

Hence, the pricing problem is

$$\min_{j=1,\ldots,m} \min_{t \in T(j)} -y_j + \sum_{i=1}^{n} t_i (p_{ij} - z_i).$$

This problem can be solved in polynomial time as follows: For each bin j, finding a packing $t \in T(j)$ of minimum reduced cost means solving a 0-1-knapsack problem with n unit size items, profit $-(p_{ij} - z_i)$ for item i, and the additional constraint that at least q_j items have to be packed into the knapsack. This problem can be solved by greedily selecting the item i with minimum value $p_{ij} - z_i$ until we have either selected B_j items, or the next item i satisfies $p_{ij} - z_i \geq 0$. If this procedure returns an infeasible packing with less than q_j items, we continue selecting the item i with minimum value $p_{ij} - z_i$ (which is now always nonnegative) until we have selected exactly q_j items. Afterwards, we can solve the pricing problem by simply comparing the best packings obtained for all bins in order to find a packing of globally minimum reduced cost. Hence, we obtain:

Theorem 1. *The linear relaxation of IP (1) can be solved in time polynomial in the encoding length of the given instance of SAP.*

4 The Randomized Rounding Procedure

We now present our randomized 3.93-approximation algorithm for SAP. In the algorithm, we first solve the linear relaxation of of IP (1) obtaining an optimal fractional solution $x \in [0,1]^{|T|}$. We then multiply all values x_t by a factor $\alpha \in [0,1]$ (which will be chosen later) and consider the resulting value $\bar{x}_t := \alpha x_t \in [0,1]$ as the probability of using packing $t \in T(j)$ for bin j. More precisely, we independently select a packing for each bin j at random, where packing $t \in T(j)$ is selected with probability $\bar{x}_t = \alpha x_t$, and with probability $1 - \sum_{t \in T(j)} \bar{x}_t$, bin j is not opened. Since we select at most one packing for each bin, the resulting vector $x^{\text{IP}} \in \{0,1\}^{|T|}$ (where $x_t^{\text{IP}} = 1$ if and only if packing t was selected) then satisfies constraint (1b), but is, in general, not a feasible solution to IP (1) since it may violate constraint (1c) (an item may be packed several times into different bins). In particular, the total number of items assigned to bins in x^{IP} may be larger than n (when counted with multiplicities). The expected profit $\mathbb{E}(\text{PROFIT}(x^{\text{IP}}))$, however, is exactly equal to $\alpha \cdot \text{PROFIT}(x)$, i.e., exactly α times the profit $\text{PROFIT}(x) =: \text{OPT}_{\text{LP}}$ of the optimal fractional solution x obtained for the linear relaxation. We note this fact for later reference:

Observation 1. The vector $x^{\text{IP}} \in \{0,1\}^{|T|}$ obtained from the randomized rounding process satisfies $\mathbb{E}(\text{PROFIT}(x^{\text{IP}})) = \alpha \cdot \text{OPT}_{\text{LP}}$.

We now show how we can turn x^{IP} into a feasible solution of IP (1) while only decreasing the expected profit by a constant factor. Our procedure works in two steps: In the first step, we discard a subset of the bins opened in x^{IP} in order to ensure that the total number of places used in the bins is at most n. In the second step, we can then replace all remaining multiply assigned items in the solution by unassigned items in order to obtain a feasible integral solution.

We start by describing the first step of the procedure. Given the vector $x^{\text{IP}} \in \{0,1\}^{|T|}$ obtained from the randomized rounding process, we consider the following instance of the 0-1-knapsack problem (0-1-KP): The objects are the packings $t \in T$ with $x_t^{\text{IP}} = 1$, i.e., the packings selected by x^{IP}. The size of object t is the number of items contained in packing t and its profit is the profit p_t of the packing. The knapsack capacity is set to n.

Assuming that the total number of places used in the bins in x^{IP} is in $[kn, (k+1)n)$ for some $k \in \mathbb{N}$, it is easy to compute an integral solution to this knapsack instance with profit at least $\frac{1}{2k+1} \cdot \text{PROFIT}(x^{\text{IP}})$: We can assign all objects fractionally to at most $(k+1)$ knapsacks of size n each such that at most k objects are fractionally assigned (cf. Figure 1). Since the size of each object is at most n, we can then remove the fractionally assigned objects from the knapsacks and put each of them into its own (additional) knapsack, which yields an integral assignment of all objects to at most $2k+1$ knapsacks. Since all objects together have total profit $\text{PROFIT}(x^{\text{IP}})$, this implies that the objects in the most profitable one among these $2k+1$ knapsacks correspond to an integral solution of the knapsack instance with profit at least $\frac{1}{2k+1} \cdot \text{PROFIT}(x^{\text{IP}})$ as desired[1] and, by choosing only the corresponding packings, we lose profit at most

$$\left(1 - \frac{1}{2k+1}\right) \cdot \text{PROFIT}(x^{\text{IP}}) = \frac{2k}{2k+1} \cdot \text{PROFIT}(x^{\text{IP}}). \tag{2}$$

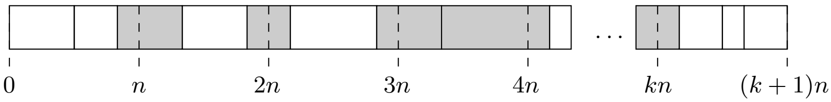

Fig. 1. Fractional assigment of the objects to $(k+1)$ knapsacks of size n each. Fractionally assigned objects are shown in grey.

In order to bound the expected loss in profit resulting from using only the packings in our solution to the knapsack instance, we now consider the probability $\Pr(k)$ that the total number of places used in the bins in x^{IP} is at least kn for each $k \in \{1, 2, \dots\}$ (if at most n places are used, we can use all packings selected by x^{IP}, so we do not lose any profit in this step). To this end, note that, by constraint (1c), the total number of places used in the optimal fractional solution x of IP (1) is at most n. Hence, since we used each packing $t \in T(j)$ with probability $\bar{x}_t = \alpha x_t$, the expected number of places used in x^{IP} is at most αn. Thus, Markov's inequality yields that

$$\Pr(k) = \Pr\left(\#(\text{places used in } x^{\text{IP}}) \geq kn\right) \leq \frac{\alpha n}{kn} = \frac{\alpha}{k} \text{ for } k \in \{1, 2, \dots\}. \tag{3}$$

In the following, $\Pr([kn, (k+1)n))$ will denote the probability that the total number of places used in the bins in x^{IP} is in $[kn, (k+1)n)$ and $\ln(\cdot)$ will denote

[1] Note that this bound on the profit of an integral solution of the knapsack instance is tight as long as $k < \frac{n}{4} - \frac{1}{2}$ as the example of $2k+1$ objects of size $\lfloor \frac{n}{2} \rfloor + 1 > \frac{n}{2}$ with unit profits shows. Hence, also computing an optimal solution for the knapsack instance (which is possible in polynomial time as the knapsack capacity n is polynomial) would not yield a better bound in general.

the natural logarithm. By (2) and (3), we then obtain that, in expectation, we lose at most the following factor times the profit of x^{IP} in the first step:

$$
\begin{aligned}
&\sum_{k=1}^{\infty} \Pr\left([kn,(k+1)n)\right)\cdot\frac{2k}{2k+1}\\
&=\sum_{k=1}^{\infty}\left(\sum_{l=k}^{\infty}\Pr\left([ln,(l+1)n)\right)-\sum_{l=k+1}^{\infty}\Pr\left([ln,(l+1)n)\right)\right)\cdot\frac{2k}{2k+1}\\
&=\sum_{k=1}^{\infty}\sum_{l=k}^{\infty}\Pr\left([ln,(l+1)n)\right)\cdot\frac{2k}{2k+1}-\sum_{k=2}^{\infty}\sum_{l=k}^{\infty}\Pr\left([ln,(l+1)n)\right)\cdot\frac{2(k-1)}{2(k-1)+1}\\
&=\sum_{l=1}^{\infty}\Pr\left([ln,(l+1)n)\right)\cdot\frac{2}{3}+\sum_{k=2}^{\infty}\sum_{l=k}^{\infty}\Pr\left([ln,(l+1)n)\right)\cdot\left(\frac{2k}{2k+1}-\frac{2(k-1)}{2(k-1)+1}\right)\\
&=\Pr(1)\cdot\frac{2}{3}+\sum_{k=2}^{\infty}\Pr(k)\cdot\left(\frac{2k}{2k+1}-\frac{2(k-1)}{2(k-1)+1}\right)\\
&=\Pr(1)\cdot\frac{2}{3}+\sum_{k=2}^{\infty}\Pr(k)\cdot\frac{2}{4k^2-1}\\
&\leq\alpha\cdot\frac{2}{3}+\sum_{k=2}^{\infty}\frac{\alpha}{k}\cdot\frac{2}{4k^2-1}\\
&=\alpha\cdot\sum_{k=1}^{\infty}\frac{2}{k(4k^2-1)}\\
&=\alpha\cdot(4\ln(2)-2)\\
&=2\alpha\cdot(2\ln(2)-1)
\end{aligned}
$$

Using Observation 1, this proves the following result:

Proposition 1. *The packings obtained after the first step contain at most n items in total and have expected profit at least $\alpha\cdot(1-2\alpha\,(2\ln(2)-1))\cdot\mathrm{OPT}_{\mathrm{LP}}$.*

In the second step of our procedure, we now have to get rid of all multiply assigned items in the solution obtained after the first step. Denoting by $j_1(i),\ldots,j_{k(i)}(i)$ the bins a multiply assigned item i is currently assigned to, we simply delete item i from all bins but the one among $j_1(i),\ldots,j_{k(i)}(i)$ in which it yields the highest profit. Doing so for all multiply assigned items yields a solution in which no item is packed more than once. The minimum quantities of the bins, however, may not be satisfied any more after deleting the multiply assigned items. But since the total number of places used in the bins after the first step was no more than the total number n of items available, we know that, for each item i that was assigned to $l\geq 2$ bins, there must be $l-1$ items that were not assigned to any bin after the first step. Hence, we can refill the $l-1$ places vacated by deleting item i from all but one bin with items that were previously unassigned, and doing so for all multiply assigned items yields a feasible integral solution to the given instance of SAP.

In order to bound the expected loss in profit resulting from the second step of our procedure, we want to bound the loss in profit resulting from deleting a single item i from all but the most profitable bin it was previously assigned to. To do so, we use that, by constraint (1c) and the scaling of the probabilities x_t given by the optimal fractional solution of IP (1) by α, the expected number of bins item i was assigned to *before* the first step of the procedure is at most α. Hence, by Markov's inequality, the probability that item i was assigned to at least k bins before the first step of the procedure can be upper bounded as

$$\Pr(i \text{ in} \geq k \text{ bins}) \leq \frac{\alpha}{k}. \tag{4}$$

Clearly, discarding a subset of the bins opened cannot increase the number of bins item i is assigned to, so inequality (4) is still valid after the first step of our procedure. Hence, denoting the probability that item i was assigned to *exactly* k bins after the first step by $\Pr(i \text{ in } k \text{ bins})$, we lose at most the following factor times the total profit obtained from all copies of item i in the solution from the first step:

$$\begin{aligned}
&\sum_{k=2}^{m} \Pr(i \text{ in } k \text{ bins}) \cdot \frac{k-1}{k} \\
&= \sum_{k=2}^{m} \left(\sum_{l=k}^{m} \Pr(i \text{ in } l \text{ bins}) - \sum_{l=k+1}^{m} \Pr(i \text{ in } l \text{ bins}) \right) \cdot \frac{k-1}{k} \\
&= \sum_{k=2}^{m} \sum_{l=k}^{m} \Pr(i \text{ in } l \text{ bins}) \cdot \frac{k-1}{k} - \sum_{k=3}^{m} \sum_{l=k}^{m} \Pr(i \text{ in } l \text{ bins}) \cdot \frac{k-2}{k-1} \\
&= \sum_{l=2}^{m} \Pr(i \text{ in } l \text{ bins}) \cdot \frac{1}{2} + \sum_{k=3}^{m} \sum_{l=k}^{m} \Pr(i \text{ in } l \text{ bins}) \cdot \left(\frac{k-1}{k} - \frac{k-2}{k-1} \right) \\
&= \Pr(i \text{ in} \geq 2 \text{ bins}) \cdot \frac{1}{2} + \sum_{k=3}^{m} \Pr(i \text{ in} \geq k \text{ bins}) \cdot \left(\frac{k-1}{k} - \frac{k-2}{k-1} \right) \\
&= \Pr(i \text{ in} \geq 2 \text{ bins}) \cdot \frac{1}{2} + \sum_{k=3}^{m} \Pr(i \text{ in} \geq k \text{ bins}) \cdot \frac{1}{k(k-1)} \\
&\leq \frac{\alpha}{2} \cdot \frac{1}{2} + \sum_{k=3}^{m} \frac{\alpha}{k} \cdot \frac{1}{k(k-1)} \\
&\leq \alpha \cdot \sum_{k=2}^{\infty} \frac{1}{k^2(k-1)} \\
&= \alpha \cdot \left(2 - \frac{\pi^2}{6} \right)
\end{aligned}$$

Together with the bound on the profit of the packings obtained after the first step given in Proposition 1, this shows the following result:

Proposition 2. *The second step of the procedure yields a feasible integral solution to the given instance of SAP with expected profit at least*

$$\left(1+\left(\frac{\pi^2}{6}-2\right)\alpha\right)\cdot\Big(1-2\alpha\big(2\ln(2)-1\big)\Big)\alpha\cdot\textsc{opt}_{\mathrm{LP}}.$$

Choosing the value α^* maximizing the expected profit in Proposition 2 (which is approximately 0.556339) yields an expected profit of at least $0.254551\cdot\textsc{opt}_{\mathrm{LP}}$. As $\textsc{opt}_{\mathrm{LP}}$ is an upper bound on the profit of the optimal integral solution of the given instance, taking the inverse of this factor and rounding up yields the following theorem:

Theorem 2. *With the right choice of α, the randomized rounding procedure yields a randomized* 3.93*-approximation algorithm for SAP.*

Using the probabilistic method (cf., for example, [4]), Theorem 2 yields an upper bound of 3.93 on the integrality gap of IP (1): Since the expected profit of the solution returned by the randomized rounding algorithm is at least $0.254551\cdot\textsc{opt}_{\mathrm{LP}}$, it follows that we obtain a feasible integral solution with profit at least $0.254551\cdot\textsc{opt}_{\mathrm{LP}}$ with positive probability. In particular, there always exists a feasible integral solution with profit at least $0.254551\cdot\textsc{opt}_{\mathrm{LP}}$, which (by again taking the inverse of this factor and rounding up) proves the following result:

Corollary 1. *The integrality gap of IP (1) is at most* 3.93.

5 Conclusion and Open Problems

In this paper, we obtained the first nontrivial approximation result for SAP by providing a randomized 3.93-approximation algorithm. We believe that the approximation factor of 3.93 obtained for our algorithm is not tight and can be slightly improved by using stronger probability bounds in some places in the analysis. A natural open question is whether a constant factor approximation for SAP can also be obtained by a deterministic algorithm. We believe that such deterministic approximation algorithms exist, but will likely require techniques different from the ones commonly used in approximation algorithms for the generalized assignment problem without minimum quantities.

References

1. Martello, S., Toth, P.: Knapsack Problems: Algorithms and Computer Implementations. John Wiley & Sons, Chichester (1990)
2. Cattrysse, D.G., Van Wassenhove, L.N.: A survey of algorithms for the generalized assignment problem. European Journal of Operational Research 60(3), 260–272 (1992)
3. Krumke, S.O., Thielen, C.: The generalized assignment problem with minimum quantities. European Journal of Operational Research 228(1), 46–55 (2013)
4. Alon, N., Spencer, J.H.: The Probabilistic Method. John Wiley & Sons (1992)

5. Pentico, D.W.: Assignment problems: A golden anniversary survey. European Journal of Operational Research 176(2), 774–793 (2007)
6. Chekuri, C., Khanna, S.: A PTAS for the multiple knapsack problem. SIAM Journal on Computing 35(3), 713–728 (2006)
7. Shmoys, D.B., Tardos, É.: An approximation algorithm for the generalized assignment problem. Mathematical Programming 62, 461–474 (1993)
8. Cohen, R., Katzir, L., Raz, D.: An efficient approximation algorithm for the generalized assignment problem. Information Processing Letters 100(4), 162–166 (2006)
9. Kellerer, H., Pferschy, U., Pisinger, D.: Knapsack Problems. Springer (2004)
10. Nemhauser, G.L., Wolsey, L.A.: Integer and Combinatorial Optimization. Wiley-Interscience, New York (1988)
11. Vazirani, V.V.: Approximation Algorithms. Springer (2001)
12. Coffman Jr., E.G., Garey, M.R., Johnson, D.S.: Approximation algorithms for bin packing: A survey. In: [23], pp. 46–93
13. Assmann, S.F.: Problems in Discrete Applied Mathematics. PhD thesis, Massachusetts Institute of Technology (1983)
14. Assmann, S.F., Johnson, D.S., Kleinman, D.J., Leung, J.Y.T.: On a dual version of the one-dimensional bin packing problem. Journal of Algorithms 5(4), 502–525 (1984)
15. Csirik, J., Johnson, D.S., Kenyon, C.: Better approximation algorithms for bin covering. In: Proceedings of the 12th ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 557–566 (2001)
16. Jansen, K., Solis-Oba, R.: An asymptotic fully polynomial time approximation scheme for bin covering. Theoretical Computer Science 306, 543–551 (2003)
17. Seedig, H.G.: Network flow optimization with minimum quantities. In: Operations Research Proceedings 2010: Selected Papers of the Annual International Conference of the German Operations Research Society, pp. 295–300. Springer (2010)
18. Krumke, S.O., Thielen, C.: Minimum cost flows with minimum quantities. Information Processing Letters 111(11), 533–537 (2011)
19. Zhu, X., Yuan, Q., Garcia-Diaz, A., Dong, L.: Minimal-cost network flow problems with variable lower bounds on arc flows. Computers and Operations Research 38(8), 1210–1218 (2011)
20. Thielen, C., Westphal, S.: Complexity and approximability of the maximum flow problem with minimum quantities. Networks (2013)
21. Mehlhorn, K., Ziegelmann, M.: Resource constrained shortest paths. In: Paterson, M. (ed.) ESA 2000. LNCS, vol. 1879, pp. 326–337. Springer, Heidelberg (2000)
22. Minoux, M.: A class of combinatorial problems with polynomially solvable large scale set covering/partitioning relaxations. RAIRO Recherche Opérationelle 21(2), 105–136 (1987)
23. Hochbaum, D.S. (ed.): Approximation Algorithms for NP-Hard Problems. PWS Publishing, Boston (1997)