A $(1-1/e)$-approximation algorithm for the maximum generalized assignment problem with fixed profits

Zeev Nutov*
The Open University of Israel
nutov@openu.ac.il

Israel Beniaminy
ClickSoftware Technologies
israel@clicksoftware.com

Raphael Yuster
University of Haifa
raphy@math.haifa.ac.il

Abstract

The *Max-Profit Generalized Assignment Problem* (Max-GAP) is: given sets J of bins and I of items, where each $j \in J$ has capacity $c(j)$ and each $i \in I$ has in bin j size $\ell(i,j)$ and profit $p(i,j)$, find a maximum profit feasible assignment. The problem admits a 1/2-approximation algorithm. Our main result is a $(1-1/e)$-approximation algorithm for *Max-GAP with fixed profits* when each $i \in I$ has a fixed profit $p(i,j) = p(i)$. We also give a fast 1/2-approximation algorithm for the *Multiple Knapsack with Assignment Restrictions* (MKAR) problem.

Key-words. Generalized assignment, Approximation algorithm.

1 Introduction

We consider the **Max-Profit Generalized Assignment Problem** (Max-GAP):
Instance: A set J of m bins (knapsacks), and a set I of n items. Each bin $j \in J$ has a capacity $c(j)$. Each item $i \in I$ has in bin j size $\ell(i,j)$ and profit $p(i,j)$.
Objective: Find a maximum profit feasible assignment.

*The Open University of Israel, nutov@openu.ac.il

To be more precise, an assignment for Max-GAP where every item is assigned to at most one bin is feasible if the capacity constraints are satisfied; namely, $\ell(A_j) = \sum_{i \in A_j} \ell(i,j) \leq c(j)$ for every bin $j \in J$ and the set A_j of items assigned to it.

A 1/2-approximation algorithm for Max-GAP was given by Chekuri and Khanna [6] based on a result of Shmoys and Tardos [10], that considered the corresponding minimization version Min-GAP: instead of profits $p(i,j)$ there are costs $w(i,j)$, and the objective is to find a feasible assignment of all items (assuming such an assignment exists) while minimizing the total cost.

Definition 1.1 *A* pseudopacking *is an assignment so that $\ell(i,j) \leq c(j)$ for each item i assigned to bin j, and that can be made feasible by removing from each bin at most one item.*

Since checking whether all items can be assigned is an NP-hard problem, Shmoys and Tardos [10] gave a bicriteria algorithm. They proved that for Min-GAP there is a polynomial time algorithm that finds a pseudopacking of cost $\leq \mathsf{opt}$, where opt denotes the optimal solution value. Using this, Chekuri and Khanna [6] proved that for Max-GAP there exists a polynomial time algorithm that returns a pseudopacking of profit $\geq \mathsf{opt}$. This implies a 1/2-approximation algorithm for Max-GAP [6]; for each overpacked bin, from the items assigned to it, remove the cheaper among: the largest item or the rest of the items. Chekuri and Khanna also showed that Max-GAP is APX-hard.

We consider *Max-GAP with fixed profits* when each item i has a fixed profit $p(i,j) = p(i)$ in every bin j, and prove:

Theorem 1.1 *Max-GAP with fixed profits admits a $(1 - 1/e)$-approximation algorithm.*

A particular case of Max-GAP with fixed profits is the *Multiple Knapsack with Assignment Restrictions* (MKAR) problem, where each bin $j \in J$ has capacity $c(j)$ and a specified set $I(j) \subseteq I$ of items that can be assigned to it, and each item i has size $\ell(i)$ and profit $p(i)$. A particular case of MKAR is when $p(i) = \ell(i)$ for all i; for this case several combinatorial 1/2-approximation algorithms are given in [5]. For the special case of MKAR with $\ell(i) \in \{1,2\}$ a 2/3-approximation algorithm is given in [1]. A particular well studied case of MKAR is the Multiple Knapsack problem, with $I(j) = I$ for all $j \in J$, for which Chekuri and Khanna [6] gave a PTAS. The difficulty of MKAR compared to Multiple Knapsack is in assigning items to "correct" bins. We give a simple and fast combinatorial 1/2-approximation algorithm for MKAR which does not require solving linear programs (as in Theorem 1.1 or as in [10]), generalizing [5].

Theorem 1.2 *MKAR has a 1/2-approximation algorithm whose running time $O(T(m,n) + m^2n^2)$, where $m = |J|$, $n = |I|$, and $T(m,n)$ is the time for computing a maximum cost*

(s,t)-*flow in a capacitated bipartite graph with parts of size* m *and* n.

Theorems 1.1 and 1.2 are proved in Sections 2 and 3, respectively.

2 A $(1-1/e)$-approximation algorithm

The proof of Theorem 1.1 follows. Let J be the set of m bins, and I be the set of n items. Recall that each bin j has capacity $c(j)$, and each item i has a bin-dependent size $\ell(i,j)$ and a global profit $p(i)$.

We use a standard LP-formulation for set-packing problems. For $S \subseteq I$ let $p(S) = \sum_{i\in S} p(i)$ and for a bin j, let $\ell(S,j) = \sum_{i\in S} \ell(i,j)$. Let $\Pi = \{(S,j) : \ell(S,j) \leq c(j)\}$. For every pair $(S,j) \in \Pi$ introduce a variable $y_{(S,j)}$ which represents the "amount of S packed in the bin j". Then integral feasible solutions to the following linear program correspond to feasible solutions to the Max-GAP with fixed profits instance.

$$\begin{aligned} \max \quad & \textstyle\sum_{(S,j)\in\Pi} p(S) y_{(S,j)} && (1)\\ \text{s.t.} \quad & \textstyle\sum_{(S,j)\in\Pi, i\in S} y_{(S,j)} \leq 1 && \forall i \in I \\ & \textstyle\sum_{(S,j)\in\Pi} y_{(S,j)} \leq 1 && \forall j \in J \\ & y_{(S,j)} \geq 0 && \forall (S,j) \in \Pi. \end{aligned}$$

The corresponding dual problem is:

$$\begin{aligned} \min \quad & \textstyle\sum_{i\in I} x_i + \sum_{j\in J} z_j && (2)\\ \text{s.t.} \quad & \textstyle\sum_{i\in S} x_i + z_j \geq p(S) && \forall (S,j) \in \Pi \\ & x_i, z_j \geq 0 && \forall i \in I, j \in J \end{aligned}$$

Note that (2) has exponential number of constraints, while (1) has exponential number of variables. However, any basic feasible solution of (1) has at most $n+m$ non-zero variables. Now, if we had a polynomial time separation oracle for (2), we could compute an optimal solution to (1) (the non-zero entries) in polynomial time, see Chapter 6 in [8]. The number of non-zero entries in such a computed solution is polynomial in $n+m$. Unfortunately, such an oracle may not exist, since the separation problem for (2) defined by a specific bin j is equivalent to the knapsack problem. To see this, introduce new variables $w_i = p(i) - x_i$ and rewrite the constraints in (2) as $w(S) = \sum_{i\in S} w_i \leq z_j$. Then, checking whether for a specific j there exists $(S,j) \in \Pi$ so that $w(S) > z_j$ is equivalent to checking whether there exists $S \subseteq I$ so that $\ell(S,j) \leq c(j)$ and $w(S) > z_j$. The latter is a knapsack problem. Since knapsack admits an FPTAS, we get an *approximate separation oracle*, which for any $\varepsilon > 0$

checks whether there exists $(S, j) \in \Pi$ so that $w(S) > z_j(1-\varepsilon)$. This implies that we can solve the following linear program in time polynomial in $1/\epsilon$ and in its size:

$$\begin{array}{lll} \min & \sum_{i\in I} x_i + \sum_{j\in J} z_j & \\ \text{s.t.} & \sum_{i\in S} x_i + z_j(1-\varepsilon) \geq p(S) & \forall (S,j) \in \Pi \\ & x_i, z_j \geq 0 & \forall i \in I, j \in J. \end{array} \tag{3}$$

Thus we can also solve the dual of (3), which is:

$$\begin{array}{lll} \max & \sum_{(S,j)\in\Pi} p(S) y_{(S,j)} & \\ \text{s.t.} & \sum_{(S,j)\in\Pi, i\in S} y_{(S,j)} \leq 1 & \forall i \in I \\ & \sum_{(S,j)\in\Pi} y_{(S,j)} \leq \frac{1}{1-\varepsilon} & \forall j \in J \\ & y_{(S,j)} \geq 0 & \forall (S,j) \in \Pi. \end{array} \tag{4}$$

Let ν and $\nu(\varepsilon)$ denote the optimal values of (1) and of (4), respectively. Clearly, $\nu(\varepsilon) \geq \nu$. Note that if $y(\varepsilon)$ is a feasible solution to (4) then $(1-\varepsilon)y(\varepsilon)$ is a feasible solution to (1). In particular, a feasible solution $y = (1-\varepsilon)y(\varepsilon)$ to (1) of value at least $(1-\varepsilon)\nu(\varepsilon) \geq (1-\varepsilon)\nu$ can be found in time polynomial in $1/\varepsilon$ and in the size of the problem. We will apply a standard randomized rounding on y to obtain an integral feasible solution $\tilde{y}$ to (1). The rounding procedure is as follows.

1. For each bin j choose randomly with distribution $y_{(S,j)}$ a unique set A_j of items assigned to bin j at this stage (possibly $A_j = \emptyset$).

2. For every item assigned to more than one bin, remove that item from all bins containing it, except for one, chosen arbitrarily.

Let $\tilde{y}$ be an integral solution derived from y by such randomized rounding, and let $\tilde{\nu} = \sum_{(S,j)\in\Pi} p(S)\tilde{y}_{(S,j)}$ be the (random variable corresponding to the) profit of $\tilde{y}$. Let P_i be the probability that item i is packed by $\tilde{y}$, that is, P_i is the probability that there exists $(S, j) \in \Pi$ with $i \in S$ and $\tilde{y}_{(S,j)} = 1$. Let $x_{ij} = \sum_{(S,j)\in\Pi, i\in S} y_{(S,j)}$ be the fraction of item i assigned to bin j by y, and let $x_i = \sum_{j\in J} x_{ij}$ be the overall fraction of item i assigned to all bins by y.

Proposition 2.1 *$P_i \geq (1 - 1/e + 1/(32m^2))x_i$ for every item $i \in I$.*

Proof: Let $i \in I$. Clearly, $P_i = 1 - \Pi_{j\in J}(1 - x_{ij})$. Let $T = \{j \in J : x_{ij} > 0\}$ be the set of bins $j \in J$ with x_{ij} being non-zero, and let $t = |T|$. Notice that $t \leq m = |J|$, and we may clearly assume that $t \geq 2$ since for $t = 1$ the statement to prove is trivial. The

minimum value of $1 - \Pi_{j \in J}(1 - x_{ij})$ is attained when $x_{ij} = x_i/t$ for every $j \in T$. Thus $P_i \geq 1 - (1 - x_i/t)^t$, and

$$\frac{P_i}{x_i} \geq \frac{1}{x_i}\left(1 - \left(1 - \frac{x_i}{t}\right)^t\right).$$

The right-hand side of the latter inequality is monotonically decreasing in x_i, so its minimum is reached when $x_i = 1$. Thus

$$\frac{P_i}{x_i} \geq \frac{1}{x_i}\left(1 - \left(1 - \frac{x_i}{t}\right)^t\right) \geq 1 - \left(1 - \frac{1}{t}\right)^t.$$

Since $t \leq m$, in order to complete the proof is suffices to show that for all $t \geq 2$, $(1 - 1/t)^t < 1/e - 1/(32t^2)$. To see this, let $\Delta(t) = 1/e - (1 - 1/t)^t$. Clearly $\Delta(t) > 0$ and $\Delta(t)$ is monotone decreasing. For $t \geq 2$

$$\begin{aligned}
\Delta(t) - \Delta(t+1) &= \left(1 - \frac{1}{t+1}\right)^{t+1} - \left(1 - \frac{1}{t}\right)^t \\
&= \left(1 - \frac{1}{t}\right)^t \left[\left(1 + \frac{1}{t^2-1}\right)^t \left(1 - \frac{1}{t+1}\right) - 1\right] \\
&> \left(1 - \frac{1}{t}\right)^t \left[\left(1 + \frac{t}{t^2-1}\right)\left(1 - \frac{1}{t+1}\right) - 1\right] \\
&= \left(1 - \frac{1}{t}\right)^t \left[\left(1 + \frac{t}{t^2-1} - \frac{1}{t+1} - \frac{t}{(t^2-1)(t+1)}\right) - 1\right] \\
&= (1 - \frac{1}{t})^t \frac{1}{(t^2-1)(t+1)} \\
&\geq \frac{1}{4(t^2-1)(t+1)}
\end{aligned}$$

In particular,

$$\Delta(t) \geq \sum_{s=t}^{\infty} \frac{1}{4(s^2-1)(s+1)} > \frac{1}{32t^2}.$$

□

Note that we used a solution y of value $\sum_{i \in I} p(i)x_i \geq (1 - \varepsilon)\nu$. Also note that since our algorithm is polynomial in $1/\varepsilon$ and the size of the input we may choose $\epsilon = 1/(32m^2)$ and our algorithm remains polynomial in the size of the input. It follows therefore that

$$\begin{aligned}
E(\tilde{\nu}) &= \sum_{i \in I} p(i)P_i \geq \left(1 - \frac{1}{e} + \frac{1}{32m^2}\right) \sum_{i \in I} p(i)x_i \\
&\geq \left(1 - \frac{1}{e} + \frac{1}{32m^2}\right)(1 - \varepsilon)\nu \\
&> \left(1 - \frac{1}{e} + \frac{1}{32m^2} - \varepsilon\right)\nu = \left(1 - \frac{1}{e}\right)\nu.
\end{aligned}$$

To complete the proof of Theorem 1.1, we show that the algorithm can be derandomized. We show how the method of conditional probabilities (see, e.g., [3]) can be used in our setting. As before, let y be a feasible solution to (1) of value at least $(1-\varepsilon)\nu(\varepsilon) \geq (1-\varepsilon)\nu$. Now, for each bin $j \in J$ let $S(j) = \{S : y_{(S,j)} > 0\}$. In case $\nu_j = \sum_{(S,j)\in\Pi} y_{(S,j)} < 1$ we also add the empty set to $S(j)$ and define $y_{(\emptyset,j)} = 1-\nu_j$. Recall that each $S(j)$ has only a polynomial size. When we preformed our randomized rounding we have selected a unique $A_j \in S(j)$ (each with its corresponding probability $y_{(S,j)}$) and proved that $E(\tilde{\nu}) = \sum_{i\in I} p(i)P_i > (1-1/e)\nu$. We shall now *deterministically* decide which $S \in S(j)$ to chose as A_j, starting sequentially from bin 1, until the last bin is reached, and A_m is selected. For $S \in S(1)$, let $E(\tilde{\nu} \mid S)$ denote the conditional expectation of $\tilde{\nu}$ *given* that we selected S as A_1. By the formula for conditional probabilities we have

$$E(\tilde{\nu}) = \sum_{S\in S(1)} E(\tilde{\nu} \mid S) y_{(S,j)}.$$

In particular, there *exists* $S \in S(1)$ for which $E(\tilde{\nu} \mid S) \geq E(\tilde{\nu})$. How do we locate this S? We first compute $E(\tilde{\nu})$ *precisely*. This can be done since $E(\tilde{\nu}) = \sum_{i\in I} p(i)P_i$ and since each P_i can be computed precisely as each x_{ij} is known. We sequentially test all $S \in S(1)$ (there are only a polynomial number of tests). For each $S \in S(1)$ we can compute *precisely* $E(\tilde{\nu} \mid S)$ because

$$E(\tilde{\nu} \mid S) = p(S) + \sum_{i\in I\setminus S} p(i)(1 - \Pi_{j\in J\setminus\{1\}}(1-x_{ij})).$$

Thus, we can deterministically locate $S \in S(1)$ for which $E(\tilde{\nu} \mid S) \geq E(\tilde{\nu})$ and define $A_1 = S$. We now continue to bins $2, 3, \ldots, m$ and do the same thing while maintaining the invariant that the conditional expectation, given the selections in the prior bins, never deceases. After selecting the set A_m in the last bin we have a deterministic selection whose profit is at least as good as the original expectation $E(\tilde{\nu})$, and hence at least $(1-1/e)\nu$. This completes the proof of Theorem 1.1. The proof also shows that the integrality gap of (1) is at least $(1-1/e)$, that is, there always exists an integral solution to (1) of value at least $(1-1/e)\nu$.

We remark that our algorithm extends to the following more general problem considered in [4]. The *Max-GAP with Color Constraints* problem is a generalization of Max-GAP obtained by assigning a color $c(i)$ to each item $i \in I$. The *color constraints* require that for each bin, the number of distinct colors of items assigned to it is at most a specified constant K.

Corollary 2.2 *Max-GAP with Color Constraints problem with fixed profits admits a* $(1-1/e)$*-approximation algorithm.*

Proof: The proof uses the same strategy as the one of Theorem 1.1. The definition of Π needs to be changed so that it includes only those sets of items that do not violate the color constraints. The approximate separation oracle for a specific constraint related to a specific bin checks all possible combinations of colors for the bin, and finds the approximate solution for the knapsack problem when the set of items under consideration is restricted to each color combination. If some color combination violates the constraint, then we have found a pair (S, j) violating the constraint. The number of combinations is at most $|I|^K$. Therefore, the approximate separation oracle is still performed in polynomial time. No other part of the proof is affected. □

3 A 1/2-approximation algorithm for MKAR

Before proving Theorem 1.1, we show that even highly restricted instances of MKAR are APX-hard. We note that the case when $\ell(i) \in \{1, 2\}$ (but bin capacities and item profits are arbitrary) was shown to be APX-hard in [2]. We prove:

Theorem 3.1 *MKAR with unit profits is APX-hard even on instances where all the bins have size* 3 *and* $\ell(i) \in \{1, 3\}$ *for all* i.

Proof: The proof is similar to the one given in [6] to show APX-hardness of some other Max-GAP instances, and is presented here only for completeness of exposition. The following problem is reduced to MKAR:

Maximum 3-Dimensional Matching (3DM):
Instance: An equitable partition X, Y, Z of a ground set V and a set-family $T \subseteq X \times Y \times Z$.
Objective: Find a subfamily $M \subseteq T$ of pairwise disjoint sets (matching) of maximal size.

Here is the reduction. Given an instance $((X, Y, Z), T)$ of 3DM with $|T| = m$ and $|X| = |Y| = |Z| = n$, create an instance of MKAR as follows. The set of bins is T, and the set of items is $V \cup U$ where U is a set of additional $m - n$ items. All bins have size 3, items in V have size 1, items in U have size 3, and all items have unit profits. Item $i \in V$ can be placed in bin $j \in T$ if the set j contains the element i. Items in U can be placed in any bin. Clearly, 3 items from V can fit in a bin if, and only if, they form a set in T. Thus bins with 3 items correspond to a matching in T. It follows therefore that if 3DM has a matching of size n, then MKAR has a solution of size $3n + (m - n) = 2n + m$.

A 3-bounded instance 3DM-3 of 3DM is one in which the number of occurrences of any element of V in T is at most 3. Kann [9] showed that there exists an $\varepsilon_0 > 0$ such that it is NP-hard to decide whether an instance of 3DM-3 has a matching of size n or if every

matching has size at most $(1-\varepsilon_0)n$. In the latter case, our MKAR instance can achieve a profit of at most $3(1-\varepsilon_0)n + 3\varepsilon_0 n + [m - (1-\varepsilon_0)n - 3\varepsilon_0 n/2] = 2n + m - \varepsilon_0 n/2$.

It follows therefore that it is NP-hard to decide whether our MKAR instance can achieve a profit of $2n+m$ or of $2n+m-\varepsilon_0 n/2$. The APX-hardness now follows from the fact that $m = O(n)$ for 3DM-3. □

The proof of Theorem 1.2 follows. Given an instance of MKAR, the *assignment graph* is a bipartite graph $G = (I + J, E)$ where $ij \in E$ if $i \in I(j)$ (assuming $\ell(i) \leq c(j)$ for every $i \in I(j)$). Consequently, $I(j)$ is the set of neighbors of j in G. Each edge $e = ij \in E$ has length $\ell_e = \ell(i)$ and profit $p_e = p(i)$. Let $\pi_e = p_e/\ell_e$. For a node v of G let $\delta(v)$ denote the set of edges incident to v. Consider the linear program:

$$\begin{array}{lll} \max & \sum_{e\in E} \pi_e x_e & \\ \text{s.t.} & \sum_{e\in\delta(j)} x_e \leq c(j) & \forall j \in B \\ & \sum_{e\in\delta(i)} x_e \leq \ell(i) & \forall i \in I \\ & x_e \geq 0 & \forall e \in E. \end{array} \tag{5}$$

Let x be a feasible solution to (5). We say that x_e (or e) is *fractional* if $0 < x_e < \ell_e$ (for $e = ij$, x_e is the amount of item i packed in bin j). Any non-fractional feasible solution to (5) bijectively corresponds to a solution to the MKAR instance, and has profit $\sum_{e\in E} \pi_e x_e$.

Lemma 3.2 *Let x be a basic feasible solution to (5). Then the set $F(x)$ of fractional edges is a forest so that in any connected component of $F(x)$ at most one leaf belongs to I. In particular, if M is a maximum matching in $F(x)$, then any non-isolated node $i \in I$ of $F(x)$ is matched by M.*

Proof: Let x be a feasible solution to (5), and let $P = (e_1, \ldots, e_\ell)$ be a path or a cycle in $F(x)$. Let $P' = \{e_1, e_3, \ldots\}$, $P'' = P - P' = \{e_2, e_4, \ldots\}$. Set $\varepsilon = \min\{\varepsilon^+, \varepsilon^-\}$ where $\varepsilon^+ = \min\{\ell_e - x_e : e \in P\}$ and $\varepsilon^- = \min\{x_e : e \in P\}$. Since all the edges in P are fractional, $\varepsilon > 0$. Let $d_e = \varepsilon$ for $e \in P'$, $d_e = -\varepsilon$ for $e \in P''$, and $d_e = 0$ otherwise. Let $x' = x + d$, $x'' = x - d$. It is easy to see that if P is a cycle, then x', x'' are feasible solutions. Since $x = (x' + x'')/2$, x cannot be basic. Thus $F(x)$ is a forest. A similar argument applies if P is a path in $F(x)$ between two leaves $u, v \in I$. In this case, it is easy to see that x', x'' satisfy the constraints of (5) for every node of G distinct from u, v. To see that this is so for u and for v, note that if $i \in I$ is a leaf of $F(x)$, then for $e \in \delta(i)$, $x_e > 0$ if, and only if, e is the unique edge of $F(x)$ incident to i. Consequently, in any connected component of $F(x)$ at most one leaf belongs to I. The second statement follows easily from the first statement. □

Corollary 3.3 *Given a feasible solution x to (5) we can find in $O(|E(G)|^2)$ time a feasible solution z such that $\pi \cdot z \geq \pi \cdot x$ and $F(z)$ is a forest so that in any connected component of $F(z)$ at most one leaf belongs to I, where $|E(G)|$ is the number of edges in G.*

Proof: For $Q \subseteq E$ denote $\pi(Q) = \sum\{\pi(e) : e \in Q\}$. Let P, P', P'' be as in the proof of Lemma 3.2. Set $Q = P'$ if $\pi(P') \geq \pi(P'')$ and $Q = P''$ otherwise, and $\varepsilon = \min\{\varepsilon^+, \varepsilon^-\}$, where $\varepsilon^+ = \min\{\ell_e - x_e : e \in Q\}$ and $\varepsilon^- = \min\{x_e : e \in P - Q\}$. Let $z_e = x_e + \varepsilon$ for $e \in Q$, $z_e = x_e - \varepsilon$ for $e \in P - Q$, and $z_e = x_e$ otherwise. Note that $|F(z)| \leq |F(x)| - 1$ (by the choice of ε), and that $\pi \cdot z \geq \pi \cdot x$. Furthermore, z is a feasible solution to (5) if x is, by an argument similar to the one as in the proof of Lemma 3.2. Thus, by repeatedly replacing x by z as above, we convert x into z as in the statement. Finding cycle/path P as above (or determining that such does not exist) can be done in $O(|E(G)|$ time, and each time P is found the number of fractional edges reduces by at least 1. The statement follows. □

Lemma 3.4 *Given an instance of MKAR, a pseudopacking of profit at least the optimal value of (5) can be computed in $O(T(m,n) + m^2n^2)$ time, where m, n and $T(m,n)$ are as in Theorem 1.2.*

Proof: Finding an optimal solution x to (5) (which may not be basic) is easily reduced to a max-cost flow problem as follows. We direct all edges in G from I to J, add a source s with edges si of the capacity $\ell(i)$ for every $i \in I$, and a sink t with edges jt of the capacity $c(j)$ for every $j \in J$. The costs are π_e for $e \in E$ and are zero otherwise. We then compute a flow f of maximum cost from s to t. It is easy to see that the restriction of f to E is an optimal solution to (5). Then we compute in $O(|E(G)|^2) = O(m^2n^2)$ time a feasible solution z as in Corollary 3.3. Finally, we compute a maximum matching M in $F(z)$ and set $x_e = \ell_e$ for every $e \in M$. It is easy to see that the resulting (possibly non-feasible) solution that has no fractional edges corresponds to a pseudopacking as required. □

As was mentioned in the Introduction, after a pseudopacking of profit at least opt is found, an assignment of profit at least $\mathsf{opt}/2$ can be found in linear time. Thus Lemma 3.4 implies Theorem 1.2.

We note that the integrality gap of (5) is 1/2 even for unit profits. Note that in the previous section we used a different LP-formulation to get a better approximation ratio.

Acknowledgment We thank an anonymous referee for useful comments.

References

[1] J. Aerts, J. Korst and F. Spieksma *Approximation of a retrieval problem for parallel disks*, Lecture Notes in Computer Science 2653 (2003), 178–188.

[2] J. Aerts, J. Korst, F. Spieksma, W. Verhaegh, and G. Woeginger, *Random redundant storage in disk arrays: complexity of retrieval problems*, IEEE Transactions on Computers, Vol. 52, no. 9 (2003), 1210–1214.

[3] N. Alon and J.H. Spencer, *The Probabilistic Method, Second Edition*, Wiley, New York, 2000.

[4] M. Dawande and J. Kalagnanam, *The Multiple Knapsack Problem with Color Constraints*, IBM Research Report RC 21128 (94508), March 24, 1998.

[5] M. Dawande, J. Kalagnanam, P. Keskinocak, R. Ravi, and F. S. Salman, *Approximation algorithms for the multiple knapsack problem with assignment restrictions*, J. of Combinatorial Optimization 4 (2000), 171–186.

[6] C. Chekuri and S. Khanna, *A PTAS for the multiple Knapsack Problem*, Proceedings of the 11^{th} annual ACM-SIAM Symposium on Discrete Algorithms (SODA) 2000, 213–222.

[7] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman, San-Francisco, 1979.

[8] M. Grötschel, L. Lovász, and A. Schrijver, *Geometric algorithms and combinatorial optimization*, Springer, 1998.

[9] V. Kann, *Maximum bounded* 3*-dimensional matching is MAX-SNP-complete*, Information Processing Letters 37 (1991), 27–35.

[10] D. B. Shmoys and E. Tardos, *An approximation algorithm for the generalized assignment problem*, Math. Programming A 62 (1993), 461–474.