

Hybrid Rounding Techniques for Knapsack Problems*

Monaldo Mastrolilli and Marcus Hutter

IDSIA, Galleria 2, 6928 Manno, Switzerland, {monaldo, marcus}@idsia.ch

May 2, 2003

Abstract

We address the classical knapsack problem and a variant in which an upper bound is imposed on the number of items that can be selected. We show that appropriate combinations of rounding techniques yield novel and powerful ways of rounding. As an application of these techniques, we present a linear-storage Polynomial Time Approximation Scheme (PTAS) and a Fully Polynomial Time Approximation Scheme (FPTAS) that compute an approximate solution, of any fixed accuracy, in linear time. This linear complexity bound gives a substantial improvement of the best previously known polynomial bounds [2].

1 Introduction

In the classical *Knapsack Problem* (KP) we have a set $N := \{1, \dots, n\}$ of items and a knapsack of limited capacity. To each item we associate a positive profit p_j and a positive weight w_j . The problem calls for selecting the set of items with maximum overall profit among those whose total weight does not exceed the knapsack capacity $c > 0$. KP has the following Integer Linear Programming (ILP) formulation:

$$\text{maximize} \quad \sum_{j \in N} p_j x_j \quad (1)$$

$$\text{subject to} \quad \sum_{j \in N} w_j x_j \leq c \quad (2)$$

$$x_j \in \{0, 1\}, \quad j \in N, \quad (3)$$

*Supported by the “Metaheuristics Network”, grant HPRN-CT-1999-00106, by Swiss National Science Foundation project 20-63733.00/1, “Resource Allocation and Scheduling in Flexible Manufacturing Systems”, and by SNF grant 2000-61847.00 to Jürgen Schmidhuber.

where each binary variable x_j , $j \in N$, is equal to 1 if and only if item j is selected. In general, we cannot take all items because the total weight of the chosen items cannot exceed the knapsack capacity c . In the sequel, without loss of generality, we assume that $\sum_{j \in N} w_j > c$ and $w_j \leq c$ for every $j \in N$.

The k -item Knapsack Problem (kKP), is a KP in which an upper bound of k is imposed on the number of items that can be selected in a solution. The problem can be formulated as (1)-(3) with the additional constraint

$$\sum_{j \in N} x_j \leq k, \quad (4)$$

with $1 \leq k \leq n$.

KP has widely been studied in the literature, see the book of Martello and Toth [9] for a comprehensive illustration of the problem. kKP is the subproblem to be solved when instances of the Cutting Stock Problem with cardinality constraints are tackled by column generation techniques. kKP also appears in processor scheduling problems on computers with k processors and shared memory. Furthermore, kKP could replace KP in the separation of cover inequalities, as outlined in [2].

Throughout our paper let OPT denote the optimal solution value to the given instance and $w(F) = \sum_{j \in F} w_j$ and $p(F) = \sum_{j \in F} p_j$, where $F \subseteq N$. An algorithm A with solution value z^A is called a $(1 - \varepsilon)$ -approximation algorithm, $\varepsilon \in (0, 1)$, if $z^A \geq (1 - \varepsilon)OPT$ holds for all problem instances. We will also call ε the *performance ratio* of A .

Known Results It is well known that KP is NP-hard but pseudopolynomially solvable through dynamic programming, and the same properties hold for kKP [2]. Basically, the developed approximation approaches for KP and kKP can be divided into three groups:

1. *Approximation algorithms.* For KP the classical $\frac{1}{2}$ -approximation algorithm (see e.g. [8]) needs only $O(n)$ running time. A performance ratio of $\frac{1}{2}$ can be obtained also for kKP by rounding the solution of the linear programming relaxation of the problem (see [2]); this algorithm can be implemented to run in linear time when the LP relaxation of kKP is solved by using the method by Megiddo and Tamir [10].
2. *Polynomial time approximation schemes (PTAS)* reach any given performance ratio and have a running time polynomial in the length of the encoded input. The best schemes currently known requiring linear space are given in Caprara et al. [2]: they yield a performance ratio of ε within $O(n^{\lceil 1/\varepsilon \rceil - 2} + n \log n)$ and $O(n^{\lceil 1/\varepsilon \rceil - 1})$ running time, for KP and kKP respectively.
3. *Fully polynomial time approximation schemes (FPTAS)* also reach any given performance ratio and have a running time polynomial in the

length of the encoded input and in the reciprocal of the performance ratio. This improvement compared to 1. and 2. is usually paid off by larger space requirements, which increases rapidly with the accuracy ε . The first FPTAS for KP was proposed by Ibarra and Kim [6], later on improved by Lawler [8] and Kellerer and Pferschy [7]. In Caprara et al. [2] it is shown that kKP admits an FPTAS that runs in $O(nk^2/\varepsilon)$ time.

New Results Rounding the input is a widely used technique to obtain polynomial time approximation schemes [4]. Among the developed rounding techniques, arithmetic or geometric rounding are the most successfully and broadly used ways of rounding to obtain a simpler instance that may be solved in polynomial time (see Sections 2.1 and 2.2 for an application of these techniques to kKP). We contribute by presenting a new technical idea. We show that appropriate combinations of arithmetic and geometric rounding techniques yield novel and powerful rounding methods. To the best of our knowledge, these techniques have never been combined together. By using the described rounding techniques, we present a PTAS for kKP requiring linear space and running time $O(n+k \cdot (1/\varepsilon)^{O(1/\varepsilon)})$. Our algorithm is clearly superior to the one in [2], and it is worth noting that the running time contains no exponent on n dependent on ε . Since KP is a special case of kKP, we also speed up the previous result for KP to $O(n \cdot (1/\varepsilon)^{O(1/\varepsilon)})$. Finally we present a faster FPTAS for kKP that runs in $O(n+k/\varepsilon^4+1/\varepsilon^5)$ time and has a bound of $O(n+1/\varepsilon^4)$ on space requirements.

2 Rounding techniques for kKP

The aim of this section is to transform any input into one with a smaller size and a simpler structure without dramatically decreasing the objective value. We discuss several transformations of the input problem. Some transformations may potentially decrease the objective function value by a factor of $1 - O(\varepsilon)$, so we can perform a constant number of them while still staying within $1 - O(\varepsilon)$ of the original optimum. Others are transformations which do not decrease the objective function value. When we describe the first type of transformation, we shall say it produces $1 - O(\varepsilon)$ *loss*, while the second produces *no loss*.

Let P^H denote the solution value obtained in $O(n)$ time by employing the $1/2$ -approximation algorithm $H^{\frac{1}{2}}$ for kKP described in [2]. In [2], it is shown that

$$2P^H \geq P^H + p_{\max} \geq OPT \geq P^H, \quad (5)$$

where $p_{\max} = \max_j p_j$.

Throughout this section we restrict our attention to feasible solutions with at most γ items, where γ is a positive integer not greater than k . The

first observation is that at most an ε -fraction of the optimal profit OPT is lost by discarding all items j where $p_j \leq \varepsilon P^H / \gamma$, since at most γ items can be selected and $P^H \leq OPT$. From now on, consider the reduced set of items with profit values greater than $\varepsilon P^H / \gamma$, with $1 - \varepsilon$ loss.

In order to reduce further the set of items, a useful insight is that when profits are identical we pick items in non-decreasing order of weight. Since the optimal profit is at most $2P^H$, for each fixed $\bar{p} \in \{p_1, \dots, p_n\}$, we can keep the first $\bar{n} = \min \left\{ \gamma, \left\lfloor \frac{2P^H}{\bar{p}} \right\rfloor \right\}$ items with the smallest weights, and discard the others with no loss. Of course, we cannot hope to obtain a smaller instance if all profits are different. In the following, we show how the number of different profits can be reduced by rounding the original profits. We revise two rounding techniques and show that an appropriate combination of both yields to a better result. We call a profit value \bar{p} *large* if $\bar{p} > \frac{2P^H}{\gamma}$, and *small* otherwise.

2.1 Arithmetic rounding

A sequence a_1, a_2, \dots is called an *arithmetic sequence* if, and only if, there is a constant d such that $a_i = a_1 + d \cdot (i - 1)$, for all integers $i \geq 1$. Let us consider the arithmetic sequence $S_a(\gamma)$ obtained by setting $a_1 = d = \varepsilon P^H / \gamma$. We transform the given instance into a more structured one by rounding each profit down to the nearest value in $S_a(\gamma)$. Since in the rounded instance the profit of each item is decreased by at most $\varepsilon P^H / \gamma$, and at most γ items can be selected, the solution value of the transformed instance potentially decreases by εP^H . Of course, by restoring the original profits we cannot decrease the objective function value, and therefore, with $1 - \varepsilon$ loss, we can assume that every possible profit is equal to $a_i = \frac{\varepsilon P^H}{\gamma} \cdot i$ for some $i \geq 1$. Furthermore, since $p_{\max} = \max_{j \in N} p_j \leq P^H$, the number of different profits is now bounded by $\left\lfloor \frac{\gamma p_{\max}}{\varepsilon P^H} \right\rfloor \leq \left\lfloor \frac{\gamma}{\varepsilon} \right\rfloor$.

The largest number n_i of items with profit a_i , for $i = 1, \dots, \left\lfloor \frac{\gamma}{\varepsilon} \right\rfloor$, that can be involved in any feasible solution is bounded by

$$n_i \leq \min \left\{ \gamma, \left\lfloor \frac{OPT}{\frac{\varepsilon P^H}{\gamma} i} \right\rfloor \right\} \leq \min \left\{ \gamma, \left\lfloor \frac{2\gamma}{\varepsilon i} \right\rfloor \right\},$$

and we can keep the first n_i items with the smallest weights, and discard the others with no loss. Now, the number of items with profit a_i is at most γ , if a_i is a small profit (i.e. when $i = 1, \dots, \left\lfloor \frac{2}{\varepsilon} \right\rfloor$), and at most $\left\lfloor \frac{2\gamma}{\varepsilon i} \right\rfloor$ otherwise ($i = \left\lfloor \frac{2}{\varepsilon} \right\rfloor + 1, \dots, \left\lfloor \frac{\gamma}{\varepsilon} \right\rfloor$). Thus, by applying the described arithmetic rounding we have at most $\lfloor 2/\varepsilon \rfloor \gamma$ items with small profits and $\sum_{i=\left\lfloor \frac{2}{\varepsilon} \right\rfloor+1}^{\left\lfloor \frac{\gamma}{\varepsilon} \right\rfloor} \left\lfloor \frac{2\gamma}{\varepsilon i} \right\rfloor$ with large profits. Recall that when a summation can be expressed as $\sum_{k=x}^y f(k)$, where $f(k)$ is a monotonically decreasing function, we can approximate it by an integral (see, e.g. [3] p. 50):

$\int_x^{y+1} f(k)dk \leq \sum_{k=x}^y f(k) \leq \int_{x-1}^y f(k)dk$. Furthermore, we are assuming that $0 < \varepsilon < 1$, and recall that $\ln(1+x) \geq x/(1+x)$, for $x > -1$. Therefore, the total number of items in the transformed instance is bounded by

$$\begin{aligned} \left\lfloor \frac{2}{\varepsilon} \right\rfloor \gamma + \sum_{i=\lfloor \frac{2}{\varepsilon} \rfloor + 1}^{\lfloor \frac{\gamma}{\varepsilon} \rfloor} \left\lfloor \frac{2\gamma}{\varepsilon i} \right\rfloor &\leq \frac{2}{\varepsilon} \gamma + \frac{2}{\varepsilon} \gamma \sum_{i=\lfloor \frac{2}{\varepsilon} \rfloor + 1}^{\lfloor \frac{\gamma}{\varepsilon} \rfloor} \frac{1}{i} \\ &\leq \frac{2\gamma}{\varepsilon} \left(1 + \int_{\frac{2}{\varepsilon}-1}^{\frac{\gamma}{\varepsilon}} \frac{di}{i}\right) = \frac{2\gamma}{\varepsilon} (1 + \ln \gamma - \ln(2 - \varepsilon)) \\ &\leq \frac{2\gamma}{\varepsilon} (1 + \ln \gamma) = O\left(\frac{\gamma}{\varepsilon} \ln \gamma\right). \end{aligned}$$

We see that by applying the described arithmetic rounding we have at most $2\gamma/\varepsilon$ items with small profits and $\frac{2\gamma}{\varepsilon} \ln \gamma$ with large profits.

A natural question is to see if the provided bound is tight. Consider an instance having γ items for each distinct profit $\frac{\varepsilon P^H}{\gamma} \cdot i$, where $i = 1, \dots, \lfloor \frac{\gamma}{\varepsilon} \rfloor$. Observe that by applying the described arithmetic rounding, we have exactly $\lfloor 2/\varepsilon \rfloor \gamma$ items with small profits and $\sum_{i=\lfloor \frac{2}{\varepsilon} \rfloor + 1}^{\lfloor \frac{\gamma}{\varepsilon} \rfloor} \lfloor \frac{2\gamma}{\varepsilon i} \rfloor$ with large profits. What remains to be shown is to bound the number of items to be $\Omega(\frac{\gamma}{\varepsilon} \ln \gamma)$:

$$\begin{aligned} \left\lfloor \frac{2}{\varepsilon} \right\rfloor \gamma + \sum_{i=\lfloor \frac{2}{\varepsilon} \rfloor + 1}^{\lfloor \frac{\gamma}{\varepsilon} \rfloor} \left\lfloor \frac{2\gamma}{\varepsilon i} \right\rfloor &\geq \frac{2}{\varepsilon} \gamma - 1 + \sum_{i=\frac{2}{\varepsilon}+1}^{\frac{\gamma}{\varepsilon}-1} \left(\frac{2\gamma}{\varepsilon i} - 1\right) \\ &\geq \frac{2}{\varepsilon} \gamma - 1 + \frac{2}{\varepsilon} \gamma \sum_{i=\frac{2}{\varepsilon}+1}^{\frac{\gamma}{\varepsilon}-1} \frac{1}{i} - \left(\frac{\gamma}{\varepsilon} - 2\right) \\ &\geq \frac{\gamma}{\varepsilon} \left(1 + 2 \int_{\frac{2}{\varepsilon}+1}^{\frac{\gamma}{\varepsilon}} \frac{di}{i}\right) = \frac{\gamma}{\varepsilon} (1 + 2 \ln \gamma - 2 \ln(2 + \varepsilon)) \\ &= \Omega\left(\frac{\gamma}{\varepsilon} \ln \gamma\right). \end{aligned}$$

2.2 Geometric rounding

A sequence a_1, a_2, \dots is called a *geometric sequence* if, and only if, there is a constant r such that $a_i = a_1 \cdot r^{i-1}$, for all integers $i \geq 1$. Let us consider the geometric sequence $S_g(\gamma)$ obtained by setting $a_1 = \varepsilon P^H / \gamma$ and $r = \frac{1}{1-\varepsilon}$. We round each profit down to the nearest value among those of $S_g(\gamma)$. Since $a_i = (1-\varepsilon)a_{i+1}$, for $i \geq 1$, each item profit is at most decreased by a factor¹ of $1-\varepsilon$, and consequently, the solution value of the transformed instance potentially decreases by the same factor of $1-\varepsilon$. Therefore, with $1-2\varepsilon$ loss,

¹This is true only for profits larger than a_1 ; recall that we are assuming, with $1-\varepsilon$ loss, that all profits are larger than a_1 .

we can assume that every possible profit is equal to $a_i = \frac{\varepsilon P^H}{\gamma} \cdot (\frac{1}{1-\varepsilon})^{i-1}$ for some $i \geq 1$. Furthermore, since $p_{\max} \leq P^H$, the number of different profits is bounded by the biggest integer β such that

$$\frac{\varepsilon P^H}{\gamma} \cdot (\frac{1}{1-\varepsilon})^{\beta-1} \leq P^H.$$

Since $\ln(\frac{1}{1-\varepsilon}) \geq \varepsilon$, we have $\beta - 1 \leq \frac{\ln(\gamma/\varepsilon)}{\ln(\frac{1}{1-\varepsilon})} \leq \frac{1}{\varepsilon} \ln \frac{\gamma}{\varepsilon}$. In any feasible solution, the largest number n_i of items with profit a_i , for $i = 1, \dots, \beta$, is bounded by

$$n_i \leq \min\left\{\gamma, \left\lfloor \frac{OPT}{\frac{\varepsilon P^H}{\gamma} \cdot (\frac{1}{1-\varepsilon})^{i-1}} \right\rfloor\right\} \leq \min\left\{\gamma, \left\lfloor \frac{2\gamma}{\varepsilon} (1-\varepsilon)^{i-1} \right\rfloor\right\},$$

and we can keep the first n_i items with the smallest weights, and discard the others with no loss. Let $\alpha = \left\lfloor \frac{\ln \lfloor 2/\varepsilon \rfloor}{\ln \frac{1}{1-\varepsilon}} \right\rfloor + 1$. Again, the number of items with profit a_i is at most γ , if a_i is a small profit (i.e. when $1 \leq i \leq \alpha$), and at most $\left\lfloor \frac{2\gamma}{\varepsilon} (1-\varepsilon)^{i-1} \right\rfloor$ otherwise ($i = \alpha + 1, \dots, \beta$). Therefore, the total number of items in the transformed instance is bounded by

$$\alpha\gamma + \sum_{i=\alpha+1}^{\beta} \left\lfloor \frac{2\gamma}{\varepsilon} (1-\varepsilon)^{i-1} \right\rfloor \leq \left(\frac{1}{\varepsilon} \ln(2/\varepsilon) + 1\right)\gamma + \frac{2\gamma}{\varepsilon} = O\left(\frac{\gamma}{\varepsilon} \ln \frac{1}{\varepsilon}\right).$$

Moreover, we can easily show that this bound is tight. Consider an instance having γ items for each distinct profit $\frac{\varepsilon P^H}{\gamma} \cdot (\frac{1}{1-\varepsilon})^{i-1}$, where $i = 1, \dots, \beta$. By applying the described geometric rounding technique, we have exactly $\alpha\gamma$ items with small profits and $\sum_{i=\alpha+1}^{\beta} \left\lfloor \frac{2\gamma}{\varepsilon} (1-\varepsilon)^{i-1} \right\rfloor$ with large profits. What remains to be shown is to bound the number of items to be $\Omega(\frac{\gamma}{\varepsilon} \ln \frac{1}{\varepsilon})$:

$$\begin{aligned} \alpha\gamma + \sum_{i=\alpha+1}^{\beta} \left\lfloor \frac{2\gamma}{\varepsilon} (1-\varepsilon)^{i-1} \right\rfloor &\geq \alpha\gamma \geq \frac{\ln \lfloor 2/\varepsilon \rfloor}{\ln \frac{1}{1-\varepsilon}} \gamma \\ &\geq \gamma \frac{1-\varepsilon}{\varepsilon} \ln \frac{2-\varepsilon}{\varepsilon} = \Omega\left(\frac{\gamma}{\varepsilon} \ln \frac{1}{\varepsilon}\right). \end{aligned}$$

We see that by applying the geometric rounding we have at most γ/ε items with large profit, while $O(\frac{\gamma}{\varepsilon} \ln \frac{1}{\varepsilon})$ items with small profits. Contrary to arithmetic rounding, the set of items that has been reduced most is the set with large profits. This suggests us to combine the described rounding techniques as described in the following subsection.

2.3 Parallel Arithmetic & Geometric rounding

We use arithmetic rounding for the set of items with small profits and geometric rounding for large items. Let us say that these two techniques are

applied in “parallel”. More formally, let us consider the *hybrid* sequence $S_{ag}(\gamma) = (a_1, a_2, \dots)$ defined by setting

$$a_i = \frac{\varepsilon P^H}{\gamma} \cdot \begin{cases} i & \text{for } i = 1, \dots, \lfloor 2/\varepsilon \rfloor, \\ \left(\frac{1}{1-\varepsilon}\right)^{\alpha+i-\lfloor 2/\varepsilon \rfloor-1} & \text{otherwise.} \end{cases}$$

We round each profit down to the nearest value among those of $S_{ag}(\gamma)$. Now, consider each set Z_i of items with the same rounded profit value a_i , and take the first

$$n_i = \begin{cases} \gamma & \text{for } i = 1, \dots, \lfloor 2/\varepsilon \rfloor, \\ \left\lfloor \frac{2\gamma}{\varepsilon} (1-\varepsilon)^{\alpha+i-\lfloor 2/\varepsilon \rfloor-1} \right\rfloor & \text{otherwise,} \end{cases}$$

items with the smallest weights we get in Z_i . Selecting the first n_i items with the smallest weights can be done in $O(|Z_i|)$ time. That is, $O(|Z_i|)$ time is sufficient to select the n_i -th item with the smallest weight (see [1]) and only $O(|Z_i|)$ comparisons are needed to extract the $n_i - 1$ items with smaller weights. Therefore the amortized time is linear.

By using the arithmetic rounding technique for small items, we have at most $2\gamma/\varepsilon$ small items with $1 - \varepsilon$ loss (see Section 2.1). While, by using the geometric rounding technique described in Section 2.2 for large items, we have at most γ/ε large items with $1 - \varepsilon$ loss. The resulting transformed instance has at most $3\gamma/\varepsilon$ items with $1 - 2\varepsilon$ loss. Furthermore, let $\psi = \beta - \alpha + \lfloor 2/\varepsilon \rfloor + 1$. Observe that the ψ -th element of $S_{ag}(\gamma)$ is larger than P^H , i.e. $a_\psi > P^H$. Consider any subset $S \subseteq N$ of items with at most γ items, and let x_i denote the total number of items from S with profit in interval $[a_i, a_{i+1})$, $i = 1, 2, \dots, \psi - 1$. Let us call vector $(x_1, x_2, \dots, x_{\psi-1})$ an *S-configuration*. It is easy to see that by using the reduced set of items it is always possible to compute a solution having the same *S-configuration*, as any set $S \subseteq N$ with γ items. Summarizing:

Lemma 1 *For any positive integer $\gamma \leq k$, it is possible to compute in linear time a reduced set $N_\gamma \subseteq N$ of items with at most $3\gamma/\varepsilon$ items, such that, for any subset $S \subseteq N$ with at most γ items, there exists a subset $S_\gamma \subseteq N_\gamma$ such that S_γ is the subset of N having the same configuration as S and with the least weights.*

Corollary 2 *For any subset $S \subseteq N$ with at most γ items, there exists a subset $S_\gamma \subseteq N_\gamma$ with $w(S_\gamma) \leq w(S)$, $|S_\gamma| = |S|$ and $p(S_\gamma) \geq p(S) - 2\varepsilon \cdot OPT$.*

3 An improved PTAS for kKP

Our PTAS for kKP improves the scheme of Caprara et al. [2], and in fact it strongly builds on their ideas. However, there are several differences where a major one is the use of two reduced sets of items instead of the entire

set N : let $\ell := \min\{\lceil 1/\varepsilon \rceil - 2, k\}$, where $\varepsilon \leq 1/2$ is an arbitrary small rational number; our algorithm uses sets N_k and N_ℓ computed by using the Arithmetic & Geometric rounding technique (see Lemma 1) when $\gamma := k$ and $\gamma := \ell$, respectively.

For any given instance of kKP, the approximation scheme performs the following five steps (S-1)-(S-5).

- (S-1) Initialize the solution A to be the empty set and set the corresponding value P^A to 0.
- (S-2) Compute the reduced sets N_k and N_ℓ .
- (S-3) Compute P^H , i.e. the solution value returned by $H^{\frac{1}{2}}$ when applied to the whole set of instances.
- (S-4) Consider each $L \subseteq N_\ell$ such that $|L| \leq \ell$. If $w(L) \leq c$, consider sequence $S_{ag}(\ell) = (a_1, a_2, \dots)$ and let h be the smallest integer such that $\min_{j \in L} p_j < a_{h+1}$ (assume $\min_{j \in L} p_j = 0$ if $L = \emptyset$). Apply algorithm $H^{\frac{1}{2}}$ to the subinstance S defined by item set $\{i \in N_k \setminus L : p_i < a_{h+1}\}$, by capacity $c - w(L)$ and by cardinality upper bound $k - \ell$. Let T and $P^H(S)$ denote the solution and the solution value returned by $H^{\frac{1}{2}}$ when applied to S , respectively. If $p(L) + P^H(S) > P^A$ let $A := L \cup T$ and $P^A := p(L) + P^H(S)$.
- (S-5) Return solution A of value P^A .

Observe that in step (S-4), subsets L are computed by considering just the items from N_ℓ . On the other hand, in step (S-4), we remark that the subinstances S are defined by using items from N_k .

3.1 Analysis of the Algorithm

Step (S-2) can be performed in $O(n)$ time by Lemma 1. Step (S-3) runs in $O(n)$ time [2]. In step (S-4) the algorithm considers $O(|N_\ell| + |N_\ell|^2 + \dots + |N_\ell|^\ell) = O(|N_\ell|^\ell)$ subsets. For each L the definition of subinstance S requires $O(|N_k| \cdot \ell)$ time. Algorithm $H^{\frac{1}{2}}$ applied to subinstance S runs in $O(|S|) = O(|N_k|)$ time [2]. By Lemma 1, $|N_k| = O(k/\varepsilon)$ and $|N_\ell| = O(\ell/\varepsilon)$. Therefore, step (S-4) is performed in $O(|N_\ell|^\ell \cdot |N_k| \cdot \ell) = O(k \cdot (\frac{\ell}{\varepsilon})^{\ell+1}) = k \cdot (1/\varepsilon)^{O(1/\varepsilon)}$. It follows that the overall running time of the algorithm is $O(n + k \cdot (1/\varepsilon)^{O(1/\varepsilon)})$, and it is not difficult to check that steps (S-1)-(S-5) require linear space. What remains to be shown is that steps (S-1)-(S-5) return a $(1 - O(\varepsilon))$ -approximate solution.

Consider sequence $S_{ag}(\ell) = (a_1, a_2, \dots)$. Let $\{j_1, \dots, j_\ell, \dots\}$ be the set of items in an optimal solution ordered so that $p_{j_1} \geq \dots \geq p_{j_\ell} \geq \dots$, and let $\lambda \leq \ell$ be the largest integer such that $p_{j_\lambda} \geq a_1$ (if there is no item in the

optimal solution with profit $\geq a_1$ then set $\lambda = 0$). Let $L^* = \{j_1, \dots, j_\lambda\}$ be the subset (possibly empty) obtained by picking the first λ items with the largest profits. Consider subinstance S^* defined by item set

$$I_{S^*} = \left\{ i \in N \setminus L^* \mid \begin{array}{l} p_i \leq \min_{j \in L^*} p_j \text{ if } L^* \neq \emptyset \\ p_i < a_1, \text{ otherwise.} \end{array} \right\},$$

by capacity $c - w(L^*)$ and by cardinality upper bound $k - \lambda$. Clearly,

$$p(L^*) + OPT_{S^*} = OPT, \quad (6)$$

where OPT_{S^*} denotes the optimal value of instance S^* . Now, consider the reduced set N_k and subinstance S_k^* defined by item set

$$I_{S_k^*} = \left\{ i \in N_k \setminus L^* \mid \begin{array}{l} p_i \leq \min_{j \in L^*} p_j \text{ if } L^* \neq \emptyset \\ p_i < a_1, \text{ otherwise.} \end{array} \right\}, \quad (7)$$

by capacity $c - w(L^*)$ and by cardinality upper bound $k - \lambda$. By Corollary 2, we have

$$OPT_{S_k^*} \geq OPT_{S^*} - 2\varepsilon OPT, \quad (8)$$

where $OPT_{S_k^*}$ denotes the optimal value of instance S_k^* .

Let us use L to denote the set of items having the same configuration as L^* and the least weights. By Lemma 1, in one of the iterations of step (S-4), set L is considered. In the remainder, let us focus on this set L , and consider the corresponding subinstance S defined in step (S-4). By Corollary 2, we have

$$p(L) \geq p(L^*) - 2\varepsilon OPT. \quad (9)$$

We need to show that the optimal solution value OPT_S of instance S cannot be smaller than $OPT_{S_k^*}$.

Lemma 3 $OPT_S \geq OPT_{S_k^*}$.

Proof. Assume L having the same configuration as L^* . Recall that the subinstance S is defined by item set $I_S = \{i \in N_k \setminus L : p_i < a_{h+1}\}$, where a_{h+1} is the term of sequence $S_{ag}(\ell) = (a_1, a_2, \dots)$ such that h is the smallest integer with $\min_{j \in L} p_j < a_{h+1}$ (see step (S-4)). On the other hand, the subinstance S_k^* is defined by item set $I_{S_k^*}$ (see (7)). If $L^* = \emptyset$ then $S = S_k^*$ and the claim follows.

Otherwise ($L^* \neq \emptyset$), since we are assuming that L has the same configuration as L^* , there are no items from L^* with profit in intervals $[a_i, a_{i+1})$, for $i < h$. Therefore, we have $\min_{j \in L^*} p_j \geq a_h$ and $\{i \in N_k \setminus L : p_i \leq a_h\} = \{i \in N_k \setminus L^* : p_i \leq a_h\}$. Furthermore, since there is at least one item from L with profit in interval $[a_h, a_{h+1})$ (recall we are assuming $L^* \neq \emptyset$), and since L^* has the same configuration as L , there exists an item from L^*

with profit $p_j < a_{h+1}$ and, therefore, $\min_{j \in L^*} p_j < a_{h+1}$. It follows that $I_{S_k^*} \subseteq \{i \in N_k \setminus L^* : p_i < a_{h+1}\}$.

By the previous arguments, the items of S_k^* , except those belonging to $A_h = \{i \in N_k \cap L : a_h \leq p_i < a_{h+1}\}$, are also items of S , i.e.,

$$I_{S_k^*} \subseteq I_S \cup A_h.$$

If there exists an optimal solution for S_k^* such that no one of the items from A_h is selected, then $OPT_S \geq OPT_{S_k^*}$, since the knapsack capacity of S_k^* is not greater than the one of S , i.e. $c - w(L) \geq c - w(L^*)$ (recall that L is the subset having the same configuration as L^* with the smallest weights).

Otherwise, let G_1 be the subset of items from A_h in an optimal solution for S_k^* , and let $g := |G_1|$. Let G_2 be any subset of $\{i \in L^* \setminus L : a_h \leq p_i < a_{h+1}\}$ containing exactly g items. It is easy to see that G_2 exists (recall that L and L^* have the same configurations and $A_h \subseteq L$). Furthermore, since $G_2 \subseteq L^*$ and $G_1 \subseteq I_{S_k^*}$, we have

$$\min_{j \in G_2} p_j \geq \max_{j \in G_1} p_j. \quad (10)$$

Observe that $w(L^*) - w(L) \geq w(G_2) - w(G_1)$. Therefore, the knapsack capacity $c - w(L)$ of S cannot be smaller than $c - w(L^*) + w(G_2) - w(G_1)$. The solution G_{12} obtained from the optimal solution for S_k^* by replacing the items from G_1 with those from G_2 , requires a knapsack of capacity bounded by $c - w(L^*) + w(G_2) - w(G_1)$. Therefore, G_{12} is a feasible solution for S since the capacity of S is greater than the capacity of S_k^* by at least $w(G_2) - w(G_1)$. Finally, from inequality (10), the solution value of G_{12} is not smaller than $OPT_{S_k^*}$ and the claim follows. ■

Let $P^H(S)$ denote the solution value returned by $H^{\frac{1}{2}}$ when applied to S . Then we have the following

Lemma 4 $p(L) + P^H(S) \geq (1 - 4\varepsilon)OPT$.

Proof. Observe that by Lemma 3 and inequality (8), we have

$$OPT_S \geq OPT_{S^*} - 2\varepsilon OPT. \quad (11)$$

We distinguish between two cases.

1. If $p(L^*) \geq (1 - \varepsilon)OPT$ then by inequalities (5), (6), (9) and (11), we

have

$$\begin{aligned} p(L) + P^H(S) &\geq p(L^*) - 2\varepsilon OPT + \frac{1}{2}OPT_S \\ &\geq (1 - \varepsilon)OPT - 2\varepsilon OPT = (1 - 3\varepsilon)OPT. \end{aligned}$$

2. If $p(L^*) < (1 - \varepsilon)OPT$ then each item profit in S^* is smaller than $\frac{(1-\varepsilon)}{\ell}OPT$. Indeed, if $\lambda = \ell$ then the smallest item profit in L^* , and hence each item profit in S^* , must be smaller than $\frac{(1-\varepsilon)}{\ell}OPT$ (otherwise $p(L^*) \geq (1 - \varepsilon)OPT$); else ($\lambda < \ell$) by definition of λ , there are at most λ items with profits not smaller than a_1 and therefore, each item profit in S^* , must be smaller than $a_1 = \frac{\varepsilon}{\ell}P^H \leq \frac{\varepsilon}{\ell}OPT \leq \frac{(1-\varepsilon)}{\ell}OPT$ (since $\varepsilon \leq 1/2$). Now, we claim that the largest profit in S is at most $\frac{(1-\varepsilon)}{\ell}OPT + \frac{\varepsilon P^H}{\ell}$. Indeed, since by definition of h we have $a_h \leq \frac{(1-\varepsilon)}{\ell}OPT \leq (1 - \varepsilon)\frac{2P^H}{\ell} \leq (\lfloor \frac{2}{\varepsilon} \rfloor - 1)\frac{\varepsilon P^H}{\ell}$, it turns out that $h \leq \lfloor \frac{2}{\varepsilon} \rfloor - 1$, and by definition of $S_{ag}(\ell)$, we have that $a_{h+1} = a_h + \frac{\varepsilon P^H}{\ell}$. Therefore, for each item j belonging to S , profit p_j is bounded by

$$p_j \leq \frac{\varepsilon}{\ell}P^H + \frac{(1 - \varepsilon)}{\ell}OPT \leq \frac{OPT}{\ell}.$$

Since $OPT_S - P^H(S) \leq \max_{j \in S} p_j$ (see inequality (5)), we have

$$\begin{aligned} p(L) + P^H(S) + \frac{OPT}{\ell} &\geq p(L) + OPT_S \\ &\geq p(L^*) + OPT_{S^*} - 4\varepsilon \cdot OPT = (1 - 4\varepsilon)OPT. \end{aligned}$$

■

By the previous lemma, steps (S-1)-(S-5) return a solution that cannot be worse than $(1 - 4\varepsilon)OPT$. Thus, we have proved the following

Theorem 5 *There is an PTAS for the k -item knapsack problem requiring linear space and $O(n + k \cdot (1/\varepsilon)^{O(1/\varepsilon)})$ time.*

To compare our algorithm with the one provided in [2] notice that the running time complexity of the latter is $O(n^{\lceil 1/\varepsilon \rceil - 1})$, whereas our scheme is linear. As in [2], our algorithm can be easily modified to deal with the *Exact k -item Knapsack Problem*, that is a kKP in which the number of items in a feasible solution must be exactly equal to k . The time and space complexities, and the analysis of the resulting algorithm are essentially the same as the one described above. Compare also with the general problem solver developed in [5], where the elimination of a multiplicative constant (here $1/\varepsilon$) led to a larger additive constant (here $(1/\varepsilon)^{O(1/\varepsilon)}$).

4 An improved FPTAS for kKP

The main goal of this section is to present a different combination of arithmetic and geometric rounding techniques. Moreover we propose an improved fully polynomial time approximation scheme that runs in $O(n + k/\varepsilon^4 + 1/\varepsilon^5)$

time. First we discuss separately the different steps in details, then we state the main algorithm and summarize the results in Section 4.2.

We start partitioning the set of items in two subsets $\mathcal{L} = \{j : p_j > \varepsilon P^H\}$ and $\mathcal{S} = \{j : p_j \leq \varepsilon P^H\}$. Let us say that \mathcal{L} is the set of *large* items, while \mathcal{S} the set of *small* items. Observe that the number of large items in any feasible solutions is not greater than $\lambda = \min\{k, \lfloor 2/\varepsilon \rfloor\}$, since $OPT \leq 2P^H$.

4.1 Dynamic programming for large items

In principle, an optimal solution could be obtained in the following way. Enumerate all different solutions for items in \mathcal{L} , i.e., consider all different sets $U \subseteq \mathcal{L}$ such that $w(U) \leq c$ and $|U| \leq k$. For each of these U , compute a set $T \subseteq \mathcal{S}$ such that $w(T) + w(U) \leq c$, $|U| + |T| \leq k$ and $p(T)$ is maximized. Select from these solutions one with the largest overall profit. One of the problems with this approach is that constructing all possible solutions for items in \mathcal{L} would require considering $n^{O(1/\varepsilon)}$ cases. To avoid the exponential dependence on $1/\varepsilon$ (our aim is to obtain a fully polynomial approximation scheme), we will not treat separately all of these solutions. We begin with the description of a basic procedure that generates a list of all “interesting” feasible combinations of profit and number of selected large items. Each such combination is represented by a pair (a, l) , for which there is a subset of items $U \subseteq \mathcal{L}$ with $p(U) = a$, $|U| = l$ and $w(U) \leq c$. Moreover $w(U)$ is the smallest attainable weight for a subset of large items with profit at least equal to a and cardinality at most l . This list of all “interesting” feasible combinations is computed by using a pseudopolynomial dynamic programming scheme. Clearly, an optimal solution can be computed by using only the subsets U of large jobs associated to each pair (a, l) . The time complexity will be then reduced, with $1 - O(\varepsilon)$ loss, by applying arithmetic and geometric rounding techniques, as described in Section 4.1.3.

Let α be the number of large items, and let β denote the number of all distinct feasible solution values obtained by considering only large items, i.e. β is the size of set

$$V = \{p(U) \mid U \subseteq \mathcal{L} \text{ and } w(U) \leq c \text{ and } |U| \leq k\}.$$

A straightforward dynamic programming recursion which has time complexity $O(\alpha\beta\lambda)$ and space complexity $O(\lambda^2\beta)$ (see [2]), can be stated as follows. Let us renumber the set of items such that the first $1, \dots, |L|$ items are large. Denote by function $g_i(a, l)$ for $i = 1, \dots, |L|$, $a \in V$, $l = 1, \dots, \lambda$, the optimal solution of the following problem:

$$g_i(a, l) = \min \sum_{j=1}^i w_j x_j : \left. \begin{array}{l} \sum_{j=1}^i p_j x_j = a; \\ \sum_{j=1}^i x_j = l; \\ x_j \in \{0, 1\}, j = 1, \dots, i. \end{array} \right\}$$

One initially sets $g_0(a, l) = +\infty$ for all $l = 0, \dots, \lambda$, $a \in V$, and then $g_0(0, 0) = 0$. Then, for $i = 1, \dots, |L|$ the entries for g_i can be computed from those of g_{i-1} by using the formula

$$g_i(a, l) = \min \left\{ \begin{array}{l} g_{i-1}(a, l), \\ g_{i-1}(a - p_i, l - 1) + w_i \quad \text{if } l > 0 \text{ and } a \geq p_i \end{array} \right\}.$$

Since $\beta = O(P^H)$ the described dynamic programming algorithm is only pseudopolynomial. In order to reduce the time complexity, we first preprocess large items by using a combination of arithmetic and geometric rounding techniques, then we apply the above dynamic programming scheme. We start analyzing the two rounding techniques separately, then we show how to combine them.

4.1.1 Geometric rounding

The time complexity of the described dynamic programming can be reduced by decreasing the number α of large items and the number β of distinct solution values.

We observed in Section 2 that if we want to reduce as much as possible the number of large items it is convenient to use geometric rounding. Consider the geometric sequence $S_g(\gamma)$ described in Section 2.2. By applying the geometric rounding technique with $\gamma = \lambda$, the number α of large items can be reduced from $O(n)$ to $O(1/\varepsilon^2)$ with $1 - \varepsilon$ loss.

The next step is to compute the number of possible solution values after geometric rounding, i.e. the cardinality β of set V after that all profit values of large items have been geometrically rounded. The main result of this section is stated as follows.

Theorem 6 *The number of solution values after geometric rounding can be exponential in $\frac{1}{\varepsilon}$.*

By the above theorem it follows that the running time of dynamic programming after geometric rounding is a constant that may depend exponentially on $\frac{1}{\varepsilon}$. Therefore, to avoid this exponential dependence on $\frac{1}{\varepsilon}$, we will look at other rounding techniques.

Proof of Theorem 6. In the remaining part of this subsection we prove Theorem 6. The goal is to derive a lower bound on the number of possible solution values after geometric rounding, i.e. a lower bound on $|V|$. Recall that we defined the geometric sequence $a_i = \frac{\varepsilon P^H}{\lambda} \left(\frac{1}{1-\varepsilon}\right)^{i-1}$ in Section 2.2, and here we assume that $\gamma = \lambda$. We focus on worst-case analysis. With this aim let us consider an instance I that after geometric rounding has at least $\lfloor P^H/a_i \rfloor$ items for each distinct profit value a_i . Moreover, we assume that $w_j = p_j$ for every $j \in \mathcal{L}$, $c = P^H$ and $k \geq 1/\varepsilon$. By definition of instance I

we see that every subset U with $p(U) < P^H$ is a feasible solution. Indeed, we have $w(U) < P^H = c$ and $|U| < p(U)/(\min_{j \in \mathcal{L}} p_j) < 1/\varepsilon \leq k$. By the previous arguments we see that $|V|$ is bounded by below by the number of solution values $y = \sum_{i=0}^{\infty} c_i a_{i+1} < P^H$ with $\mathbf{c} = (c_0, c_1, \dots) \in \mathbb{N}_0^\infty$, where \mathbb{N}_0^∞ is the set of sequences with non-negative integer components. Inserting a_i and $\varepsilon = \frac{\varepsilon'}{1+\varepsilon'}$ this is equivalent to

$$\sum_{i=0}^{\infty} c_i \varepsilon' (1 + \varepsilon')^i < \lambda(1 + \varepsilon'). \quad (12)$$

Clearly

$$\sum_{i=0}^{\infty} c_i \varepsilon' (1 + \varepsilon')^i < 1. \quad (13)$$

implies (12), since $\lambda(1 + \varepsilon') > 1$.

For simplicity of notation we replace ε' with ε , and we focus on the cardinality of sets of the form

$$R_\varepsilon^d := \left\{ y < 1 : y = \sum_{i=0}^{d-1} c_i \varepsilon (1 + \varepsilon)^i, \quad \mathbf{c} \in \mathbb{N}_0^d \right\},$$

where \mathbb{N}_0^d is the set of d -dimensional vectors $\mathbf{c} = (c_0, c_1, \dots, c_{d-1})$ with non-negative integer components. It is more easy to find lower bounds on the set of vectors

$$S_\varepsilon^d := \left\{ \mathbf{c} \in \mathbb{N}_0^d : \sum_{i=0}^{d-1} c_i \varepsilon (1 + \varepsilon)^i < 1 \right\} \quad (14)$$

itself. To consider $|S_\varepsilon^d|$ instead of $|R_\varepsilon^d|$ is justified by the following Lemma.

Lemma 7 *For rational and for transcendental $\varepsilon > 0$, the sets R_ε^d and S_ε^d have the same cardinality, i.e. the mapping $f : S_\varepsilon^d \rightarrow R_\varepsilon^d$ with $f(\mathbf{c}) = \sum_{i=0}^{d-1} c_i \varepsilon (1 + \varepsilon)^i$ is one-to-one and onto.*

Proof for transcendental ε .

$$y = y' \Leftrightarrow \sum_{i=0}^{d-1} b_i x^i = 0 \quad \text{with} \quad b_i := c_i - c'_i \in \mathbb{Z} \quad \text{and} \quad x := 1 + \varepsilon$$

A real number is said to be transcendental if it is not the root of a polynomial with integer coefficients. For transcendental ε also x is transcendental, which implies $b_i \equiv 0$. Hence, $y = y'$ implies $\mathbf{c} = \mathbf{c}'$. Obviously $\mathbf{c} = \mathbf{c}'$ implies $y = y'$. This proves that S_ε^d has the same cardinality as R_ε^d for transcendental ε .

Proof for rational ε . Assume by contradiction that there are two different vectors $\mathbf{c} \neq \mathbf{c}' \in \mathbb{N}_0^d$ with same solution value $y = y'$. Let n be the

largest index i such that $c_i \neq c'_i$. Furthermore, let $\varepsilon = \frac{p}{q} - 1 > 0$ be rational with $p > q \in \mathbb{N}$ having no common factors. With $\mathbf{b} := \mathbf{c} - \mathbf{c}'$ we have

$$\begin{aligned} 0 &= [y - y'] = \sum_{i=0}^{d-1} b_i \varepsilon (1 + \varepsilon)^i \\ &= \varepsilon \sum_{i=0}^n b_i \frac{p^i}{q^i} = \frac{\varepsilon}{q^n} \left[b_n p^n + q \overbrace{\sum_{i=0}^{n-1} [b_i p^i q^{n-i-1}]}^{\text{integer}} \right] \end{aligned}$$

Since the last term $q \sum[\dots]$ is a multiple of q , $y - y'$ can only be zero if also $b_n p^n$ is a multiple of q . With p also p^n has no common factor with q , hence b_n must itself be a multiple of q . The sum in (14) can only be less than 1 if each term is less than 1, i.e. $c_i \varepsilon (1 + \varepsilon)^i < 1$. This implies

$$c_i < \varepsilon^{-1} (1 + \varepsilon)^{-i} \leq \varepsilon^{-1} \quad \text{for all } i. \quad (15)$$

Together we get

$$0 \leq c_n^{(')} < \frac{1}{\varepsilon} = \frac{q}{p - q} < q \Rightarrow |b_n| = |c_n - c'_n| < q \Rightarrow b_n = 0 \Rightarrow c_n = c'_n$$

which contradicts our assumption $c_n \neq c'_n$. Hence, $y = y'$ implies $\mathbf{c} = \mathbf{c}'$. Again, that $\mathbf{c} = \mathbf{c}'$ implies $y = y'$ is obvious. This shows that S_ε has the same cardinality as R_ε for rational ε . ■

We don't know whether Lemma 7 also holds for algebraic ε . The following Lemma lower bounds S_ε^∞ .

Lemma 8 $|S_\varepsilon^\infty| \geq C e^{B/\varepsilon}$ with $B = 0.3172\dots$ and $C = 0.3200\dots$

Proof. From (15) we see that all c_i are zero for too large i ($c_i = 0 \forall i \geq d_{max} := \lceil \frac{\ln(1/\varepsilon)}{\ln(1+\varepsilon)} \rceil$). This shows that $|S_\varepsilon^1| \leq |S_\varepsilon^2| \leq \dots \leq |S_\varepsilon^{d_{max}}| = |S_\varepsilon^{d_{max}+1}| = \dots = |S_\varepsilon^\infty|$. The main idea in the following is to relate S_ε^d to the volume of a d -dimensional simplex with volume larger than $C e^{B/\varepsilon}$ for suitable d .

We define a d -dimensional subset $U_\varepsilon^d \subset \mathbb{R}^d$, which is the disjoint union of unit cubes $[c_0, c_0 + 1) \times \dots \times [c_{d-1}, c_{d-1} + 1)$ for every $\mathbf{c} \in S_\varepsilon^d$. This set can be represented in the following form

$$U_\varepsilon^d := \left\{ \mathbf{r} \in [0, \infty)^d : \sum_{i=0}^{d-1} [r_i] \varepsilon (1 + \varepsilon)^i < 1 \right\}$$

The volume $\text{Vol}(U_\varepsilon^d)$ coincides with the cardinality of set S_ε^d since each point in S_ε^d corresponds to exactly one unit cube in U_ε^d , each having volume 1. Furthermore, let us define the d -dimensional (irregular) tetrahedron

$$T_\varepsilon^d := \left\{ \mathbf{r} \in [0, \infty)^d : \sum_{i=0}^{d-1} r_i \varepsilon (1 + \varepsilon)^i < 1 \right\}$$

Obviously $T_\varepsilon^d \subseteq U_\varepsilon^d$, since $\lfloor r_i \rfloor \leq r_i$. So we have $|S_\varepsilon^\infty| \geq |S_\varepsilon^d| = \text{Vol}(U_\varepsilon^d) \geq \text{Vol}(T_\varepsilon^d)$. The tetrahedron T_ε^d is orthogonal at the vertex $\mathbf{r} = 0$. The edges $\mathbf{r} = (0, \dots, 0, r_i, 0, \dots, 0)$ have lengths $[\varepsilon(1 + \varepsilon)^i]^{-1}$, $i = 0 \dots d - 1$. Hence, the volume of the tetrahedron is

$$\begin{aligned} \text{Vol}(T_\varepsilon^d) &= \frac{1}{d!} \prod_{i=0}^{d-1} [\varepsilon(1 + \varepsilon)^i]^{-1} = [d! \varepsilon^d (1 + \varepsilon)^{d(d-1)/2}]^{-1} \\ &\geq e^{[-(d\varepsilon) \ln(d\varepsilon) - \frac{1}{2}(d\varepsilon)^2]/\varepsilon} = e^{f(d\varepsilon)/\varepsilon} \end{aligned}$$

with $f(x) := -x \ln x - \frac{1}{2}x^2$. In the inequality we replaced $d - 1$ by d and used $d! \leq d^d$ and $1 + \varepsilon \leq e^\varepsilon$. The best bound is found by maximizing $e^{f(d\varepsilon)/\varepsilon}$ w.r.t. d , or equivalently by maximizing $f(x)$ w.r.t. x . We have $-f'(A) = \ln A + 1 + A = 0$ for $A = 0.2784\dots$. Hence, d should be chosen as $\frac{A}{\varepsilon}$, but since d is integer we have to round somehow, for instance $d = \lfloor \frac{A}{\varepsilon} \rfloor$. Note that $|S_\varepsilon^d|$ increases with d , but our approximation becomes crude for d near d_{max} . This is the reason why the maximizing d is less than d_{max} . For small ε we have $f(\varepsilon \lfloor \frac{A}{\varepsilon} \rfloor) \approx f(\varepsilon \frac{A}{\varepsilon}) = f(A) = -A \ln A - \frac{1}{2}A^2 = A(1 + \frac{1}{2}A) =: B = 0.3172\dots$ with corrections of order $O(\varepsilon)$. This establishes an asymptotic bound $\sim e^{B/\varepsilon}$. More exactly, one can show that $f(\varepsilon \lfloor \frac{A}{\varepsilon} \rfloor) \geq f(A)(1 - \frac{\varepsilon}{A})$ for all ε . This yields the bound

$$|S_\varepsilon^\infty| \geq \max_d \text{Vol}(T_\varepsilon^d) \geq e^{f(A)(1 - \frac{\varepsilon}{A})/\varepsilon} = Ce^{B/\varepsilon} \quad \text{with} \quad C = e^{-B/A} = 0.3200\dots$$

■

The coefficient B can be improved to 0.7279... for sufficiently small ε by using the more accurate Stirling approximation for $d!$.

Using Lemma 7 and 8 it is now easy to lower bound the number of possible solution values for geometric profit distribution. From Lemma 8 we know that (13) has at least $Ce^{B/\varepsilon}$ solution vectors \mathbf{c} and from Lemma 7 that (12) has at least $Ce^{B/\varepsilon}$ solution values y for rational ε , and the proof of Theorem 6 follows.

4.1.2 Arithmetic rounding

Alternatively, we may think to apply arithmetic rounding to the set of large items. Let us consider the arithmetic sequence $S_a(\gamma)$ described in Section 2.1. By applying the arithmetic rounding technique with $\gamma = \lambda$, we observe the number of large items can be reduced to be bounded by $O(\frac{1}{\varepsilon^2} \ln \frac{1}{\varepsilon})$ with $1 - \varepsilon$ loss. Moreover each element of set V is equal to $\frac{\varepsilon P^H}{\lambda} i$ for some $i = \lambda, \lambda + 1, \dots, 2 \lfloor \lambda/\varepsilon \rfloor$. It follows that the size of set V is bounded by $O(1/\varepsilon^2)$, and the overall time of the dynamic programming algorithm is now $O(\frac{1}{\varepsilon^5} \ln \frac{1}{\varepsilon})$. We see that in comparison to the geometric rounding and although the number of large items is larger, the arithmetic rounding technique is able to reduce much more the size of set V . However and again, we can take advantage from both techniques by combining them as described in the following.

4.1.3 Serial Geometric & Arithmetic rounding

We first apply geometric rounding with $1 - \varepsilon$ loss. This reduces the number of large items to be bounded by $O(1/\varepsilon^2)$. Then, with $1 - \varepsilon$ loss, we apply arithmetic rounding on the reduced set of large items. Clearly the latter does not increase the number of items and each profit value is now equal to $\frac{\varepsilon P^H}{\lambda} i$ for some $i = \lambda, \lambda + 1, \dots, 2 \lfloor \lambda/\varepsilon \rfloor$. By using this set of items with profits rounded by using geometric first and arithmetic rounding then, the size of set V has a bound of $O(1/\varepsilon^2)$, and the overall time of the dynamic programming algorithm is $O(1/\varepsilon^5)$. We call this combination a *Serial Geometric & Arithmetic rounding technique*.

4.2 Adding small items

In the following we show how to add the small items. First, with $1 - 2\varepsilon$ loss, we reduce the number of small items to be $O(k/\varepsilon)$ by using the Parallel Arithmetic & Geometric rounding (see Section 2.3). Then, for each pair (a, l) in the final list, fill in the remaining knapsack capacity $c - g_{|L|}(a, l)$ with at most $k - l$ small items, by using algorithm $H^{\frac{1}{2}}$ for kKP [2]. These small items yield total profit $P^H(c - g_{|L|}(a, l), k - l)$. By inequality (5) and by definition of small items, we have

$$P^H(c - g_{|L|}(a, l), k - l) + \varepsilon P^H \geq OPT(c - g_{|L|}(a, l), k - l), \quad (16)$$

where $OPT(c - g_{|L|}(a, l), k - l)$ is the optimal solution value obtained by using at most $k - l$ small items and knapsack capacity $c - g_{|L|}(a, l)$. The approximate solution, a combination of large and small items, is chosen to yield profit P , where

$$P = \max_{(a, l)} \{a + P^H(c - g_{|L|}(a, l), k - l)\}$$

By inequality (16) and since our algorithm considers all the “interesting” pairs (a, l) with $1 - O(\varepsilon)$ loss, it is easy to verify that P is $1 - O(\varepsilon)$ times the optimal solution.

To summarize, the steps of the FPTAS are as follows.

- (S-1) Partition the set of items into “large” and “small”. Apply the Serial Geometric & Arithmetic rounding technique to the set of large items. Apply the Parallel Arithmetic & Geometric rounding technique to the set of small items.
- (S-2) Solve for the “large” items using dynamic programming: generate a list of all “interesting” feasible combinations (a, l) of profit a and number l of selected large items.
- (S-3) For each pair (a, l) in the final list, fill in the knapsack by applying algorithm $H^{\frac{1}{2}}$ with the reduced set small items.

(S-4) Return the best found solution.

Step (S-1) can be performed in $O(n)$ time. Step (S-2) takes $O(1/\varepsilon^5)$ time. Algorithm $H^{\frac{1}{2}}$ applied to the reduced set of small items runs in $O(k/\varepsilon)$ time [2]. In step (S-3) the algorithm considers $O(1/\varepsilon^3)$ pairs, for each one performing operations that require $O(k/\varepsilon)$ time. It follows that the overall running time of the algorithm is $O(n + k/\varepsilon^4 + 1/\varepsilon^5)$. The space complexity has a bound of $O(n + 1/\varepsilon^4)$, since the space required by the dynamic programming is $O(\lambda^2\beta)$ where $\lambda = O(1/\varepsilon)$ and $\beta = O(1/\varepsilon^2)$.

Theorem 9 *There is a fully polynomial time approximation scheme for the k -item knapsack problem requiring $O(n + k/\varepsilon^4 + 1/\varepsilon^5)$ time and $O(n + 1/\varepsilon^4)$ space.*

Acknowledgments. Thanks are due to Klaus Jansen for introducing us to the k -item Knapsack Problem. We are grateful to the referees who pointed out some mistakes in the early version of this paper.

References

- [1] M. Blum, R. W. Floyd, V. Pratt, R. Rivest, and R. Tarjan. Time bounds for selection. *Journal of Computer and System Sciences*, 7:448–461, 1973.
- [2] A. Caprara, H. Kellerer, U. Pferschy, and D. Pisinger. Approximation algorithms for knapsack problems with cardinality constraints. *European Journal of Operational Research*, 123:333–345, 2000.
- [3] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to algorithms*. MIT Press and McGraw-Hill Book Company, 6th edition, 1992.
- [4] D. Hochbaum, editor. *Approximation Algorithms for NP-hard Problems*. ITP, 1995.
- [5] M. Hutter. The fastest and shortest algorithm for all well-defined problems. *International Journal of Foundations of Computer Science*, 13(3):431–443, 2002.
- [6] O. H. Ibarra and C. E. Kim. Fast approximation algorithms for the knapsack and sum of subset problems. *J. Assoc. Comput. Mach.*, 22:463–468, 1975.
- [7] H. Kellerer and U. Pferschy. A new fully polynomial approximation scheme for the knapsack problem. *APPROX'98*, LNCS 1444:123–134, 1998.

- [8] E. L. Lawler. Fast approximation algorithms for knapsack problems. *Proc. 18th Ann. Symp. on Foundations of Computer Science*, pages 206–218, 1977.
- [9] S. Martello and P. Toth. *Knapsack Problems*. Wiley, 1990.
- [10] N. Megiddo and A. Tamir. Linear time algorithms for some separable quadratic programming problems. *Operations Research Letters*, 13:203–211, 1993.