d-dimensional Knapsack in the Streaming Model

Sumit Ganguly[1] and Christian Sohler[2]

[1] Indian Institute of Technology, Kanpur, India
[2] Technical University Dortmund, Dortmund, Germany

Abstract. We study the d-dimensional knapsack problem in the data streaming model. The knapsack is modelled as a d-dimensional integer vector of capacities. For simplicity, we assume that the input is scaled such that all capacities are 1. There is an input stream of n items, each item is modelled as a d-dimensional integer column of non-negative integer weights and a scalar profit. The input instance has to be processed in an online fashion using sub-linear space. After the items have arrived, an approximation for the cost of an optimal solution as well as a template for an approximate solution is output.

Our algorithm achieves an approximation ratio $(2(\frac{1}{2}+\sqrt{2d+\frac{1}{4}}))^{-1}$ using space $O(2^{O(d)} \cdot \log^{d+1} d \cdot \log^{d+1} \Delta \cdot \log n)$ bits, where $\{\frac{1}{\Delta}, \frac{2}{\Delta}, \ldots, 1\}$, $\Delta \geq 2$ is the set of possible profits and weights in any dimension, and P is the ratio between the minimum and maximum profit. We also show that any data streaming algorithm for the $t(t-1)$-dimensional knapsack problem that uses space $o(\sqrt{\Delta}/t^2)$ cannot achieve an approximation ratio that is better than $1/t$. Thus, even using space Δ^γ, for $\gamma < 1/2$, i.e. space polynomial in Δ, will not help to break the $1/t \approx 1/\sqrt{d}$ barrier in the approximation ratio.

1 Introduction

The 0/1 knapsack problem is a popular and well-studied combinatorial problem with applications in many different areas. Its basic form is as follows. Given n items numbered $i = 1, 2, \ldots, n$ and their weights w_i and profits p_i, find a subset of the items with maximum profit whose sum of weights does not exceed a given sack size R. The problem is well-known to be an NP-hard problem [6] with a classical FPTAS approximation algorithm by Ibarra and Kim [3].

A well-studied generalization is the d-dimensional knapsack problem, whose input is the set of items indexed by $\{1, 2, \ldots, n\}$, where the ith item is associated with (a) a non-negative d-dimensional vector $A_i = [A_{1,i}, \ldots, A_{d,i}]^T$ denoting the weight of the item along each of the d dimensions, and, (b) a profit p_i. The knapsack dimensions is given by the column vector R with R_s being the sack dimension along dimension s. The problem may be specified as follows.

$$\max_{S \subset \{1,2,\ldots,n\}} \sum_{j \in S} p_j$$
$$\text{subject to } \sum_{j \in S} A_{s,j} \leq R_s, \quad s = 1, 2, \ldots, d \ .$$

We consider the above problem in the data stream setting. For simplicity, we assume that the knapsack capacities are scaled to be 1. This is equivalent to assuming that the knapsack capacities are given as input to the algorithm instead of being read from the stream. In this setting, the items arrive in a sequence whose entries are of the form (item, profit, d-dimensional column of weights)$=$ (i, p_i, A_i). The profits and weights are chosen from $\{\frac{1}{\Delta}, \frac{2}{\Delta}, \ldots, 1\}$, $\Delta \geq 2$. The algorithm may only use a small amount of space, say $s(n, d, \Delta)$ bits, for storing the input. In the streaming scenario we aim at $s(n, d, \Delta)$ being sublinear or even polylogarithmic in the input length. Each item is processed in an online fashion. After all the items have been processed, the streaming algorithm may use its $s(n, d, \Delta)$-bit state to report a template for a packing of the items into the knapsack. The following constraints must be satisfied: (1) Any template packing of items reported by the algorithm must be feasible for the original instance, and, (2) the corresponding profit reported must not be higher than the sum of the profits of the items in the sack.

The knapsack problem in the data stream setting is substantially different than the online resource knapsack problem with resource augmentations, studied by Iwama and Taketomi [4] and by Iwama and Zhang [5]. In the online knapsack problem, each item may be seen only once, and a decision regarding whether to include it in the knapsack must be made. Additionally, items arriving later may cause a currently included item in the sack to be evicted; evicted items are not recoverable. Iwama and Zhang show that the problem is not approximable (i.e., ratio is ∞). However, if one allows resource augmentation, that is, the algorithm is allowed to store a set of items whose weight is at most $\alpha \geq 1$ knapsacks, then, a greedy algorithm attains a $\alpha - 1$ approximation ratio and this is optimal [5].

The main difference between the online resource augmented version and the streaming version is that the online version allows a set of items to be stored as long as its weight is at most αR. Since items may have small weight, it is possible that $\Theta(n)$ items are stored, which would not be allowed in the data stream setting.

Contributions and Overview. In this work, we study deterministic solutions to the d-dimensional knapsack problem in the streaming model. We assume that the input is scaled such that the sack capacities are 1. We show that for $\epsilon = O(1/\log(d))$, there is an algorithm that gives a $(2(\frac{1}{2} + \sqrt{2d + \frac{1}{4}}))^{-1}$-approximate packing using space $O(2^{O(d)}\epsilon^{-(d+1)}(\log^{d+1}\Delta)(\log n))$, where weights and profits are taken from $\{\frac{1}{\Delta}, \frac{2}{\Delta}, \ldots, 1\}$, $\Delta \geq 2$. Further, we show that for any $d = t(t-1)$, any algorithm that uses $o(\Delta/d)$ space has approximation ratio no better than $1/t$. Thus, our technique yields an approximation ratio that is within a small constant factor of the best possible. In fact, even using space *polynomial* in Δ will not help to break the $1/t \approx 1/\sqrt{d}$ barrier in the approximation ratio.

Our technique is as follows. We compress the input instance by rounding up the weights[3] and rounding down the profits, respectively, to the nearest power

[3] The actual scheme for rounding up weights is slightly more involved.

of $1+\epsilon$. The input instance is now approximated as a $d+1$-dimensional array H, where there are d co-ordinates for each of the weight dimensions and one profit coordinate. An entry $H[w_1, \ldots, w_d, p]$ in this array is the count of the number of items that have profit p and weights $w_1, \ldots, w_d$ respectively in the d-dimensions, after the rounding operation.

Our approach for the analysis is to take any given feasible solution F for the original instance and show that it can be packed into at most $2(\frac{1}{2} + \sqrt{2d + \frac{1}{4}})$ sacks using the modified weights. This is done using a 2-level hierarchical packing method. The first level is obtained as the color classes of an optimal coloring of a certain hypergraph. In the second stage, we use the probabilistic method to prove that the items of each of the color classes can be packed into 2 sacks. We show that the first level coloring has at most $\frac{1}{2} + \sqrt{2d + \frac{1}{4}}$ color classes. Since each such class is packed using at most 2 sacks, we obtain a $2(\frac{1}{2} + \sqrt{2d + \frac{1}{4}})$-sack packing of the given feasible solution F. Therefore, one among these packings has cost at least $1/(2(\frac{1}{2} + \sqrt{2d + \frac{1}{4}})$ of the original feasible solution F. By choosing the feasible solution to be the optimal solution, we obtain a $1/(2(\frac{1}{2} + \sqrt{2d + \frac{1}{4}})$-approximate solution for the modified weights instance.

Our lower bound is obtained as a reduction from the t-party set disjointness problem in one-way communication complexity.

2 Compressed representation

An instance $\mathit{\Pi} = (A, p)$ of the d-dimensional knapsack problem is a $d \times n$-dimensional weights matrix A and an n-dimensional row vector p. To simplify notation, we will assume that the item weights along each dimension, namely the $A_{i,j}$'s are from $\{\frac{1}{\Delta}, \frac{1}{\Delta}, \ldots, 1\}$, $\Delta \geq 2$. The knapsack is assumed to have unit capacity in each of its dimensions. Thus our problem is the following.

$$\max_{F \subset \{1,2,\ldots,n\}} \sum_{j \in F} p_j \quad \text{subject to } \sum_{j \in F} A_{ij} \leq 1, \quad i = 1, 2, \ldots d \ .$$

We will assume that $a = 1/\Delta$ is a lower bound on the smallest weight along any dimension. In order to represent the input instance using compact space, we discretize the weights and profits as follows. The interval $[a, 1/2]$ is divided into bins using a logarithmic scale:

$$(a, a(1+\epsilon)], (a(1+\epsilon), a(1+\epsilon)^2], \ldots, (a(1+\epsilon)^{j_0}, 1/2] \ .$$

The last interval ends at $1/2$. For $x \in [a, 1/2]$, we map x to the right end of the interval in powers of $(1+\epsilon)$ that contains x; however, we never cross $1/2$. Call this mapping $x \mapsto up_{a,\epsilon}(x)$, that is,

$$up_{a,\epsilon}(x) = \min(a(1+\epsilon)^{\lceil \log_{1+\epsilon}(x/a) \rceil}, 1/2), \qquad a \leq x \leq 1/2 \ .$$

This gives $\lceil \log_{1+\epsilon} 1/(2a) \rceil$ distinct discrete weights. Define the mapping $x \mapsto dn_{a,\epsilon}(x)$, where, $dn_{a,\epsilon}(x)$ is the left end of the interval in the above list of intervals that contains x, that is,

$$dn_{a,\epsilon}(x) = a(1+\epsilon)^{\lfloor \log_{1+\epsilon}(x/a) \rfloor}, \qquad a \leq x \leq 1/2 \ .$$

The mapping for the interval $[1/2, 1-a]$ is obtained by dividing the interval backwards, that is, $x \mapsto up_{a,\epsilon}(x)$, where,

$$up_{a,\epsilon}(x) = 1 - dn_{a,\epsilon}(1-x), \qquad 1/2 < x < 1-a \ .$$

In the interval $[a, 1/2]$, $up_{a,\epsilon}(x)$ rounds x upwards to the nearest value of the form $a(1+\epsilon)^j$.

$$a \leq up_{a,\epsilon}(x) \leq 1/2, \quad x \in [a, 1/2] \text{ and} \tag{1}$$

$$0 \leq up_{a,\epsilon}(x) - x \leq \epsilon x, \qquad \text{for } x \in [a, 1/2] \tag{2}$$

In the interval $[1/2, 1]$, $up_{a,\epsilon}(x)$ rounds x upwards to $1 - dn_{a,\epsilon}(1-x)$. Therefore,

$$(1+\epsilon)/2 \leq up_{a,\epsilon}(x) \leq 1-a, \quad x \in (1/2, 1-a], \text{ and} \tag{3}$$

$$0 \leq up_{a,\epsilon}(x) - x \leq \epsilon(1-x), \qquad \text{for } x \in (1/2, 1-a] \ . \tag{4}$$

A round-down discretization is performed for profits by mapping p_i to $dn_{p_{\min},\epsilon}(p_i)$, for the entire range of profit values, where, $p_{\min}$ is a lower bound on the smallest profit and ϵ is a parameter. This ensures that $0 \leq p_j - dn_{p_{\min},\epsilon}(p_j) \leq \epsilon p_j$, which is sufficient for our purposes.

Compressed instance. Let A' be the $d \times n$ matrix obtained by mapping each entry A_{ij} to $up_{a,\epsilon}(A_{ij})$, for $1 \leq i \leq d, 1 \leq j \leq n$. We represent A' as a histogram H in $d+1$-dimensions as follows, where, the first d dimensions are the weights and the last dimension is the profit. H contains $2\lceil \log_{1+\epsilon}(1/2a) \rceil + 2$ entries in each of first d dimensions and $\lceil \log_{1+\epsilon}(\Delta) \rceil + 1$ entries for each the profit dimension, where, P is an upper bound on profit of items. Each cell of H is initialized to 0. When an item with profit p_j and weight column A_j appears, we map A_j to the vector

$$A_j \mapsto A'_j = up_{a,\epsilon}(A_j) = [up_{a,\epsilon}(A_{1,j}), up_{a,\epsilon}(A_{2,j}), \ldots, up_{a,\epsilon}(A_{d,j})]^T$$

and the profit p_j to $dn_{p_{\min},\epsilon}(p_j)$. The histogram entry corresponding to $H[A'_j, p'_j]$ is incremented by 1.

Space requirement. Each entry of the histogram stores a count of the number of items with the same rounded weights and profit. It suffices to use $\log n$ bits for each entry. Thus, the histogram requires space

$$O((\lceil \log_{1+\epsilon}(2/a) \rceil + 2)^d (\log_{1+\epsilon}(\Delta) + 1)(\log n)) = O(2^{O(d)} \epsilon^{-d+1} \log^{d+1}(\Delta)(\log n)) \text{ bits}$$

where, n is the number of items and $\Delta \geq 2$.

A simple but important property of the rounding procedure is that, (a) a feasible solution for the modified (i.e., the rounded) instance is a feasible solution for the original instance–this is a consequence of the rounding up of the weights, and, (b) the profit of a feasible solution of the modified instance is at most the profit of the same solution for the original instance–this is a consequence of rounding down of the profits. The profit of a feasible solution of the modified instance is at least $1/(1+\epsilon)$ of the profit of the same solution for the original instance.

Definition 1. *Let $\Pi = (A, p)$ be an instance of the d-dimensional knapsack problem. A pair-wise non-intersecting family of subsets $S_1, \ldots, S_k$ is called a feasible packing using k knapsacks or, in short, a feasible k-sack solution, if*

$$\sum_{l \in S_j} A_{i,l} \leq 1, \qquad \text{for each } j = 1, 2, \ldots, k \text{ and } 1 \leq i \leq d \ .$$

We use feasible k-sack solutions as follows. Given an instance Π of a knapsack problem, let $\mathrm{OPT}(\Pi)$ denote the optimal profit feasible.

Lemma 1. *Let g be a function that maps an instance Π of the knapsack to another instance $g(\Pi)$ that does not change the profit vector p but may change the weight matrix A to A'. Suppose that for every instance Π and every feasible solution F of Π, there is a feasible k-sack solution for $g(\Pi)$ whose union of sets is F. Then, $\mathrm{OPT}(g(\Pi)) \geq \mathrm{OPT}(\Pi)/k$, for all instances Π.*

Proof. There is a k-sack feasible solution for $g(\Pi)$ whose union is $\mathrm{OPT}(\Pi)$. One of these k-sacks has profit at least $\mathrm{OPT}(\Pi)/k$. Thus $\mathrm{OPT}(g(\Pi)) \geq \mathrm{OPT}(\Pi)/k$.

Lemma 1 guides our approach towards obtaining a bound of k on the approximation ratio of the original versus the compressed instance. The bound is obtained by showing that for each feasible solution of the original instance there is a k-sack feasible solution of the compressed instance. It then follows that the optimal solution for the compressed instance is $1/k$-approximation. The problem now reduces to making k as small as possible.

3 Conflict Hypergraph and its Coloring

Given an instance Π of the knapsack problem, we denote by Π' the instance that replaces all input weights $A_{t,j}$ by $A'_{t,j} := up_{a,\epsilon}(A_{t,j})$ and profits p_j by $p'_j := dn_{p_{\min},\epsilon}(p_j)$. Let F be an arbitrary feasbile solution for Π. We define a conflict hypergraph that captures all sets of items of F that violate the sack size in any dimension for instance Π'.

Definition 2. *The hypergraph $H_F = (V, E_F)$ with $V := F$ and $E_F := \{h \subseteq V : \exists s, 1 \leq s \leq d, \sum_{j \in h} A'_{s,j} > 1\}$ is called* conflict hypergraph *of a feasible solution F. An edge h has label s, if $\sum_{j \in h} A'_{s,j} > 1$. Note that an edge can have different labels.*

A hypergraph k-coloring is an assignment $\chi : V \to \{1, \ldots, k\}$ of the vertices to one of the k colors. We call the sets $\chi^{-1}(i)$ the color classes of the coloring. A hyperedge h is called monochromatic (under a k-coloring χ), if there exists a color $c, 1 \leq c \leq k$ such that $\chi(v) = c$ for all $v \in h$. A coloring is called *proper*, if there are no monochromatic edges. A hypergraph is called k-colorable, if it has a proper k-coloring. A hypergraph has chromatic number k, if it is k-colorable but not $k-1$-colorable. We can now formulate a relationship between the error introduced by our rounding procedure and the problem of coloring the conflict hypergraph.

Lemma 2. *Let F be a feasible solution for $\Pi = (A, p)$. A feasible k-sack solution $S'_1, \ldots, S'_k$ for $\Pi' = (A', p')$ with $F = \bigcup_{i \in \{1,\ldots,k\}} S'_i$ exists, iff H_F is k-colorable.*

Proof. Let $S'_1, \ldots, S'_k$ be a feasible k-sack solution for $\Pi' = (A', p')$. We define a k-coloring χ for H_F by setting $\chi^{-1}(j) = S'_j$ for $1 \leq j \leq k$. The coloring is well-defined since by definition of a k-sack solution the sets S'_j do not intersect and since $F = \bigcup_{j \in \{1,\ldots,k\}} S'_j$. By the definition of a feasible k-sack solution we have that $\sum_{l \in S'_j} A'_{i,l} \leq 1, \quad$ for each $j = 1, 2, \ldots, k$ and $1 \leq i \leq d$. Now assume that there is a hyperedge h and a color c with $\chi(v) = c$ for all $v \in h$. Then we have that $v \in S'_c$ for all $v \in h$ and so $\sum_{l \in h} A'_{i,l} \leq \sum_{l \in S'_c} A'_{i,l} \leq 1$ for $1 \leq i \leq d$. But this is a contradiction to the fact that $h \in E_F$, which states that for some dimension s we have $\sum_{l \in h} A'_{s,l} > 1$.

Now let us assume that there is a proper coloring of H_F. Then we define $S'_j := \chi^{-1}(j)$. By definition of a proper coloring, there is no edge $h \subseteq S'_j$. Hence, $\sum_{l \in S'_j} A'_{i,l} \leq 1$ for each $1 \leq j \leq k$ and $1 \leq i \leq d$. Thus $S_1, \ldots, S_k$ is a feasible k-sack solution. □

Our approach will be to show that for every feasible solution F, the conflict hypergraph H_F has chromatic number at most $k := 2(\frac{1}{2} + \sqrt{2d + \frac{1}{4}})$. By the above lemma this implies that a k-sack feasible solution exists. By Lemma 1 this will give a $1/k$-approximate solution.

We first give a folklore result that any hypergraph with E edges can be colored using $\frac{1}{2} + \sqrt{2|E| + \frac{1}{4}}$ colors. The result follows from the observation that a coloring with minimum number of colors must have a hyperedge for every pair of colors. The proof for the graph version (see, for example, [2], page 124) extends immediately to hypergraphs.

Lemma 3 (folklore). *Let $H = (V, E)$ be a hypergraph. Then, H is $\frac{1}{2} + \sqrt{2|E| + \frac{1}{4}}$-colorable.* □

Lemma 3 will prove useful in obtaining desirable packings. However, an immediate application of Lemma 3 is not useful, since, the number of conflict edges could be exponential in the number of vertices. Thus, the number k of sacks required to obtain a k-feasible packing by Lemma 3 may be exponential. It is therefore necessary to apply a two step approach to construct a proper

$(2(\frac{1}{2}+\sqrt{2d+\frac{1}{4}}))$-coloring of H_F. Our first step will be to construct a restricted conflict graph H'_F over vertex set F and obtain a $\frac{1}{2}+\sqrt{2d+\frac{1}{4}}$-coloring of H'_F. This coloring is later refined to a $(2(\frac{1}{2}+\sqrt{2d+\frac{1}{4}}))$-coloring of H_F.

Definition 3. *Let*

$$Z_{F,s} = \bigcap_{g \in E_F \,:\, g \text{ has label } s} g \ .$$

The hypergraph $H'_F = (F, E'_F)$ with $E'_F := \{Z_{F,s} : |Z_{F,s}| > 1\}$ is called restricted conflict hypergraph.

An immediate motivation for Definition 3 is to reduce the number of hyperedges. Corresponding to any feasible solution F of the original instance, by construction, H'_F has at most one hyperedge per dimension. Hence, $|E'_F| \leq d$, and by Lemma 3, there exists a $\frac{1}{2}+\sqrt{2d+\frac{1}{4}}$-coloring χ of H'_F. However, not all conflicts in H_F are reflected in H'_F and thus, there may be monochromatic edges in H_F under the coloring χ. However, an edge with label s can only be monochromatic, if $|Z_{F,s}| \leq 1$.

Therefore we revise our strategy to a 2-level *hierarchical packing*. Given a feasible packing of the original instance, we first obtain a decomposition of the vertices into color classes that are a proper coloring of the restricted conflict hypergraph H'_F. In the second stage, we take the items in each of the color classes obtained and show that these items can be packed into 2 sacks. In order to prove this, we only have to show that for every dimension s with $|Z_{F,s}| \leq 1$ the items can be packed into two sacks. We use the probabilistic method to show this, namely, we show that a random packing is a feasible packing with constant probability. This implies that there exists a refined coloring that has no monochromatic edges.

From now on, let us assume that we are given a subset S of items that forms a color class of the restricted conflict hypergraph H'_F corresponding to some given feasible packing F of the original instance of the knapsack problem. By construction, we only have to consider dimensions s with $|Z_{F,s}| = 0$ or 1. Clearly, in the worst case we have $S = F$ and so we will show that even in this case, our random packing will succeed with large probability. We will first analyze consequences of $|Z_{F,s}|$ being 0 or 1. To simplify notation, let $A_{i,J} := \sum_{j \in J} A_{i,j}$.

Lemma 4. *Let $\epsilon < 1/3$. Suppose that we are given a feasible solution F for the original instance of the knapsack problem and a dimension $s \in \{1, 2, \ldots, d\}$ such that $|Z_{F,s}| = 1$ and $Z_{F,s} = \{u_s\}$. Let $g = \{u_s\} \cup I$ and $h = \{u_s\} \cup J$ are hyperedges in the conflict hypergraph H_F corresponding to F with label s.*

1. *If $A_{s,u_s} > 1/2$ then, $A_{s,I \cap J} > \frac{1-3\epsilon}{1+\epsilon}(1 - A_{s,u_s})$.*
2. *If $A_{s,u_s} \leq 1/2$, then, $A_{s,I \cap J} > \frac{1-\epsilon}{1+\epsilon} - A_{s,u_s}$.*

Proof. Case 1: $A_{s,u_s} > 1/2$. Since, g is not feasible for the modified weights, $A'_{s,g} > 1$. So,

$$1 < A'_{s,g} = A'_{s,u_s} + A'_{s,I} \leq \epsilon(1 - A_{s,u_s}) + A_{s,u_s} + (1+\epsilon)A_{s,I}\,.$$

Therefore, we get,

$$(1+\epsilon)A_{s,I} > (1-\epsilon)\beta_s, \text{ where, } \beta_s = (1 - A_{s,u_s})\ .$$

Applying the above inequality to the hyperedge h, we obtain similarly that $(1+\epsilon)A_{s,J} > (1-\epsilon)\beta_s$. Adding, we have,

$$(1+\epsilon)(A_{s,I} + A_{s,J}) = (1+\epsilon)(A_{s,I\cap J} + A_{s,I\cup J}) > 2(1-\epsilon)\beta_s\ .$$

However, since F is a feasible set, the set of elements $\{u_s\} \cup I \cup J$ fit in the sack along dimension s. Thus,

$$A_{s,u_s} + A_{s,I\cup J} \leq 1 \text{ or, } A_{s,I\cup J} \leq 1 - A_{s,u_s} = \beta_s$$

and therefore,

$$(1+\epsilon)(A_{s,I\cap J} + \beta_s) > 2(1-\epsilon)\beta_s$$

or,

$$A_{s,I\cap J} > \frac{2(1-\epsilon)\beta_s}{1+\epsilon} - \beta_s = \frac{(1-3\epsilon)\beta_s}{1+\epsilon}\ .$$

Case 2: $A_{s,u_s} \leq 1/2$. We have

$$1 < A'_{s,g} = A'_{s,u_s} + A'_{s,I} \leq (1+\epsilon)A_{s,u_s} + (1+\epsilon)A_{s,I}, \text{ and}$$
$$1 < A'_{s,h} = A'_{s,u_s} + A'_{s,J} \leq (1+\epsilon)A_{s,u_s} + (1+\epsilon)A_{s,J}$$

Adding, we obtain

$$\begin{aligned} 2 &< 2(1+\epsilon)A_{s,u_s} + (1+\epsilon)(A_{s,I\cup J} + A_{s,I\cap J}) \\ &= (1+\epsilon)(A_{s,u_s} + A_{s,I\cap J}) + (1+\epsilon)(A_{s,u_s} + A_{s,I\cup J})\ . \end{aligned}$$

Since F is a feasible packing, the elements $\{s\} \cup I \cup J$ all fit in the sack in terms of their original weights, that is, $A_{s,u_s} + A_{s,I\cup J} \leq 1$. Thus,

$$2 < (1+\epsilon)(A_{s,u_s} + A_{s,I\cap J}) + (1+\epsilon), \text{ or, } \qquad A_{s,I\cap J} > \frac{1-\epsilon}{1+\epsilon} - A_{s,u_s}\ . \quad \square$$

Lemma 5. *Let $\epsilon < 1/3$ and F be a feasible solution for the original instance of the knapsack problem. Let s be a dimension $s \in \{1, 2, \ldots, d\}$ such that $|Z_{F,s}| = 1$ and $Z_{F,s} = \{u_s\}$. Suppose j is a member of some hyperedge in H_F with label s. Then, the following holds.*

1. *For $j \neq u_s$ and $A_{s,u_s} > 1/2$, $A_{s,j} \leq (1 - A_{s,u_s})(4\epsilon)/(1+\epsilon)$.*
2. *For $j \neq u_s$ and $A_{s,u_s} \leq 1/2$, $A_{s,j} \leq 2\epsilon/(1+\epsilon)$.*

If j is not a member of any hyperedge in H_F with label s, then, $A_{s,j} \leq \epsilon/(1+\epsilon)$.

Proof. Let $\beta_s = 1 - A_{s,u_s}$. We will consider two cases with regard to an item $j \neq u_s$; Case 1 when an item j is a member of some hyperedge in H_F with label s, and, Case 2 when an item j is not a member of any hyperedge in H_F labeled s.

Case 1 : Suppose $j \neq u_s$ and j is a member of some hyperedge $\{u_s\} \cup I$ in H_F. Since, $j \notin Z_s$, there exists another hyperedge $h = \{u_s\} \cup J$ in H_U such that $j \notin h$. We get $A_{s,j} \leq \beta_s - A_{s,I \cap J}$.

Case 1.1 : $A_{s,u_s} > 1/2$. By Lemma 4 part (1), $A_{s,I\cap J} \geq \beta_s(1-3\epsilon)/(1+\epsilon)$. Thus,

$$A_{s,j} \leq \beta_s - \beta_s \frac{1-3\epsilon}{1+\epsilon} = \frac{4\epsilon\beta_s}{1+\epsilon} \ .$$

Case 1.2 : $A_{s,u_s} \leq 1/2$. By Lemma 4 part (2), $A_{s,I\cap J} \geq \frac{1-\epsilon}{1+\epsilon} - A_{s,u_s}$. Therefore,

$$A_{s,j} \leq 1 - A_{s,u_s} - \frac{1-\epsilon}{1+\epsilon} + A_{s,u_s} = \frac{2\epsilon}{1+\epsilon} \ .$$

Case 2 : Suppose j is not a member of any hyperedge in H_F with label s. Then, for any hyperedge g with label s (and there is at least one since $|Z_s| = 1$),

$$1 < A'_{s,g} \leq (1+\epsilon)A_{s,g} \leq (1+\epsilon)(1 - A_{s,j})$$

or, $A_{s,j} < \frac{\epsilon}{1+\epsilon}$. □

Lemma 5 implies that for the case when $|Z_{F,s}| = 1$, all elements except the largest element are of size $O(\epsilon)$. This follows from Lemma 5 by noting that $A_{s,j} \leq (1 - A_{s,u_s})(4\epsilon)/(1+\epsilon) \leq 4\epsilon/(1+\epsilon)$, since, $0 \leq 1 - A_{s,u_s} \leq 1$.

Lemma 6. *Given a feasible solution F for the original instance of the knapsack problem and a dimension $s \in \{1, 2, \ldots, d\}$ such that $|Z_{F,s}| = 0$. Then, either there are no conflicting edges along dimension s, or, all items have weight at most $\epsilon/(1+\epsilon)$ along dimension s.*

Proof. If there are no conflicting edges in H_F with label s then there is nothing to prove. So suppose there exists at least one conflicting edge g in H_F with label s.

Case 1: Suppose $j \in g$. Since $Z_{F,s} = \phi$, it follows that there is at least one other conflicting edge $h \in H_F$ such that $j \notin h$. Therefore,

$$A'_{s,h} > 1, \text{ or, } 1 < A'_{s,h} \leq (1+\epsilon)A_{s,h} \leq (1+\epsilon)(1 - A_{s,j}) \ .$$

Simplifying, we obtain $A_{s,j} \leq \frac{\epsilon}{1+\epsilon}$.

Case 2: Suppose j does not appear in any hyperedge of H_F with label s, then, the argument of Lemma 5 can be used to show that $A_{s,j} \leq \epsilon/(1+\epsilon)$. □

Packing Items in a Color Class

In this section, we consider the remaining portion of the strategy of the 2-level hierarchical packing outlined above. In the hierarchical packing strategy, given a feasible solution F for the original knapsack instance, we first form the conflict hypergraph H_F and then derive the restricted conflict hypergraph H'_F from it

as explained earlier. The vertex set F of the restricted conflict hypergraph H'_F is partitioned into the color classes of a proper coloring. The final step is to pack the items comprising each color class using as few sacks as possible. In this section, we present this step.

We will pack the items in an independent set of H'_F using a random strategy. The random packing strategy picks items at random with probability p and attempts to place them in the sack. If the sack overflows, a new sack is opened, and the process continues until there are no more items to be packed.

Lemma 7. *Let $p = 1/2$ and $\epsilon \leq \min(1/24, 1/(64\log(4d)))$ and F be a feasible solution to the original knapsack instance. Let S be a set of items that forms an color class of the restricted conflict hypergraph H'_F. Then, the probability that the sack overflows along any given dimension is at most $1/(4d)$.*

Proof. Consider a random packing that chooses each item with probability p. Define an indicator variable x_j to be 1 if item j is selected and 0 otherwise. Let u_t be an item with the maximum weight along dimension t, that is $A_{t,u_t} = \max_i A_{t,i}$. Let $\beta'_t = 1 - A'_{t,u_t}$ and for $j \neq u_t$, let

$$w'_{t,j} = \frac{A'_{t,j}}{\max_{k\neq u_t} A'_{t,k}} x_j, \qquad j \in \{1, 2, \ldots, n\} - \{u_t\}$$

and let $X_t = \sum_{j\neq u_t} w'_{t,j}$. Thus, $\mathsf{E}[X_t] = \sum_{j\neq u_t} \frac{A'_{s,j}}{\max_{k\neq u_t} A'_{t,k}} p$

Case 1.1: $|Z_{F,t}| = 1$ and $A_{t,u_t} > 1/2$. By Lemma 5, if $|Z_{F,t}| = 1$ then, $\max_{k\neq u_t} A'_{t,k} \leq 4\epsilon(1 - A_{t,u_t})/(1+\epsilon)$. We wish to obtain an upper bound for the probability that the items selected do not fit into the sack along dimension t, while leaving room for u_t. A sufficient condition for this is

$$\sum_{j\neq u_t} A'_{t,j} x_j \leq \beta'_t, \text{ or, equivalently, } X_t \leq \frac{\beta'_t}{\max_{k\neq u_t} A'_{t,k}} \ .$$

$\mathsf{Pr}\{\text{Overflow along dimension } t\} = \mathsf{Pr}\left\{X_t > \frac{\beta'_t}{\max_{k\neq u_t} A'_{t,k}}\right\}$. By Hoeffding's bound applied to the sum X_t of random variables $w'_{t,j}$, with $p \geq 1/2$, by Lemma 5 this is

$$\mathsf{Pr}\left\{X_t > \frac{\beta'_t}{\max_{k\neq u_t} A'_{t,k}}\right\} \leq \exp\left\{-(1/6)\frac{((1-\epsilon)-(1+\epsilon)p)^2}{\epsilon(1+(1+\epsilon)p)}\right\} \ . \qquad (5)$$

The details of the application of Hoeffding's inequality are presented in Appendix A.

Case 1.2: $|Z_{F,t}| = 1$ and $A_{t,u_t} \leq 1/2$. By Lemma 5, $\max_{k\neq u_t} A'_{t,k} \leq (1+\epsilon)A_{t,k} \leq 2\epsilon$.

$$\mathsf{Pr}\left\{X_t > \frac{\beta'_t}{\max_{k\neq u_t} A'_{t,k}}\right\} < \exp\left\{-(1/3)\frac{((1-\epsilon)-(1+\epsilon)p)^2}{\epsilon(1+(1+\epsilon)p)}\right\} \ . \qquad (6)$$

Equation (6) is derived in Appendix A.

Case 2: $|Z_{F,t}| = 0$. By Lemma 6, if $|Z_{F,t}| = 0$, then, $\max_k A'_{t,k} \leq (1+\epsilon)\max_k A_{t,k} \leq \epsilon$. Applying Hoeffding's bound yields,

$$\mathsf{Pr}\left\{X_t > \frac{\beta'_t}{\max_{k\neq u_t} A'_{t,k}}\right\} < \exp\left\{-(2/3)\frac{((1-\epsilon)-(1+\epsilon)p)^2}{\epsilon(1+(1+\epsilon)p)}\right\} \quad . \tag{7}$$

Combining (5), (6) and (7), we have under all cases,

$$\mathsf{Pr}\left\{X_t > \frac{\beta'_t}{\max_{k\neq u_t} A'_{t,k}}\right\} \leq \exp\left\{-(1/6)\frac{((1-\epsilon)-(1+\epsilon)p)^2}{\epsilon(1+(1+\epsilon)p)}\right\} \quad . \tag{8}$$

Therefore, $\mathsf{Pr}\left\{X_t > \frac{\beta'_t}{\max_{k\neq u_t} A'_{t,k}}\right\} \leq 1/(4d)$ provided

$$\frac{((1-\epsilon)-(1+\epsilon)p)^2}{\epsilon(1+(1+\epsilon)p)} > 6\log(4d) \tag{9}$$

Equation (9) is satisfied if $p = 1/2$ and $\epsilon = \min(1/24, 1/(64\log(4d)))$. □

We can now use Lemma 7, to prove that a 2-sack packing of the items in each color class can be obtained.

Lemma 8. *Let $p = 1/2$ and $\epsilon = \min(1/24, 1/(64\log(4d)))$ and F be a feasible solution to the original knapsack instance. Let S be a set of items that forms a color class of the restricted conflict hypergraph H'_F. Then, there exists a 2-sack feasible packing of the items in S.*

Proof. Suppose we sample items with probability 1/2. In this case, the probability that an item is not sampled is also 1/2. Thus, we can also apply the previous analysis (Lemma 7) to the set of items that were not selected in the sample. The packing of this set of items succeeds with probability $1 - d/(4d)$. Thus, by the union bound, the probability that for every dimension and both the sets of selected and not selected items are feasible is at least $1 - (2d)/(4d) = 1/2 > 0$ by the union bound. Thus a 2-sack feasible packing of the items in S exists. □

We can now state the resulting property of our hierarchical packing analysis.

Theorem 1. *For $\epsilon \leq \min(1/24, 1/(64\log(4d)))$, our algorithm computes a rounded instance such that this instance has a $(2(\frac{1}{2}+\sqrt{2d+\frac{1}{4}}))^{-1}$-approximate solution. Our rounded instance is stored in a histogram that requires $O(2^{O(d)}\log^{d+1}(d)\log^{d+1}(\Delta)(\log n))$ bits of space, for some constant c.*

Proof. The algorithm for filling the knapsack is as follows. We round up the weights as explained in Section 2 and round down the profits. In this manner we create a histogram of multi-dimensional weights. The size of this histogram is $O(2^{O(d)}(\epsilon^{-(d+1)})\log^d(\Delta)\log n)$ bits. From the histogram we can obtain a solution using exhaustive search.

The approximation factor is derived as follows. Let F be any feasible solution for the original instance. Let H_F be the conflict hypergraph and let H'_F denote the restricted conflict hypergraph (Definition 3. We then color H'_F using the smallest number of colors. Lemma 3 shows that this can be done using at most $\frac{1}{2} + \sqrt{2d + \frac{1}{4}}$ colors. Since, there are at most d edges in H'_F, we obtain at most $\frac{1}{2} + \sqrt{2d + \frac{1}{4}}$ color classes with no hyperedges from H'_F.

By Lemma 8 each color class S can be packed using $t = 2$ sacks, where, $\epsilon = O(1/(\log(d))$. This means that the set of items in F can be packed into at most $2(\frac{1}{2} + \sqrt{2d + \frac{1}{4}})$ sacks. By Lemma 1, this implies that there is a $1/(t(\frac{1}{2} + \sqrt{2d + \frac{1}{4}}))$-approximation solution. The final two statements of the theorem are special cases obtained from the corresponding special cases of Lemma 8. □

4 Lower Bounds: Space versus Approximation Ratio

In this section, we present a lower bound on space versus approximation ratio of streaming algorithms for the d-dimensional knapsack problem.

Theorem 2. *For $t \geq 1$, any streaming algorithm for the $t(t-1)$-dimensional knapsack problem with items from the universe $\{\frac{1}{\Delta}, \frac{2}{\Delta}, \ldots, 1\}$ that uses $o(\sqrt{\Delta}/t^2)$ bits has an approximation ratio of at most $1/t$.*

Let t be a positive integer such that $d = t(t-1)/2$. We reduce the t-party set disjointness problem to $2d$-dimensional knapsack problem. An instance of the t-party set disjointness problem consists of subsets $S_1, \ldots, S_t$ of $[n]$ provided to each of t players. It is given that the sets are either pair-wise disjoint, or, there is exactly one element in the common intersection. The players follow a pre-specified communication protocol at the end of which a designated player can distinguish between the two kinds of input. If the protocol is randomized, then, the final answer must be correct with probability say $7/8$. The communication complexity of a protocol is the maximum over all legal inputs, of the total number of bits communicated during the execution of the protocol. The communication complexity of the problem is the complexity of the best possible protocol for this problem. It was shown by Chakrabarti, Khot and Sun [1] that the randomized communication complexity of this problem is $\Omega(n/(t \log t))$. Assuming the one-way communication model, where, the players communicate in a certain order, that is, player 1 sends to player 2, player 2 to player 3 and so on, the communication complexity is $\Omega(n/t)$ [1].

Proof. Consider an input instance of the t-player set disjointness problem, namely, t subsets $S_1, \ldots, S_t$ of the universe $\{1, 2, \ldots, n\}$ that is provided to each of the t-parties, together with the promise that either, (a) the sets are pair-wise disjoint, or, (b) they have exactly one common intersection element and are otherwise pair-wise disjoint. We view the set $\{1, \ldots, t(t-1)\}$ as being isomorphic to the set of triples $\{(a, b, 0/1)\}$, where $1 \leq a < b \leq t$. Let $d = t(t-1)$. For each

$c = 1, 2, \ldots, t$, player c maps each element x of S_c to a d-dimensional vector denoted by $x^{(c)}$ and whose coordinates are referred to as

$$[x^{(c)}_{a,b,e}], \quad 1 \leq a < b \leq t \text{ and } e \in \{0, 1\} \ .$$

The vector $x^{(c)}$ is constructed as follows.

1. $x^{(c)}_{c,b,0} = 2x, x^{(c)}_{c,b,1} = 2(4n - x)$
2. $x^{(c)}_{a,c,0} = 2(4n - x), x^{(c)}_{a,c,1} = 2x$
3. $x^{(c)}_{a,b,0} = x^{(c)}_{a,b,1} = \epsilon$, if $a \neq c$ and $b \neq c$.

ϵ is chosen to be a sufficiently small number, say $1/(2n)$, so that $n\epsilon < 1$. Suppose that the knapsack size is $8n + 1$ along each dimension.

Case: non-empty intersection. Suppose there is a common element x in each of the S_c's. Then, $x^{(a)}_{a,b,0} = 2x, x^{(b)}_{a,b,0} = 2(4n-x)$ and $x^{(t)}_{a,b,0} = \epsilon$, for all $t \neq a, t \neq b$. So the sum of coordinate $(a, b, 0)$ of the d-dimensional vectors corresponding to x across the t parties is

$$2x + 2(4n - x) + (n - 2)\epsilon < 8n + 1 \ .$$

Similarly, the coordinate $(a, b, 1)$ of the sum of the d-dimensional vectors corresponding to x across the t parties is less than $8n + 1$. That is, the vectors $\{x^{(1)}, \ldots, x^{(t)}\}$ form a feasible packing into one knapsack.

Case: empty intersection. Choose a pair of distinct elements $x \in S_c$ and $y \in S_{c'}$, and suppose that $c < c'$. Then,

$$x^{(c)}_{c,c',0} + y^{(c')}_{c,c',0} = 2x + 2(4n - y)$$
$$x^{(c)}_{c,c',1} + y^{(c)}_{c,c',1} = 2(4n - x) + 2y$$

Both $2x + 2(4n - y)$ and $2(4n - x) + 2y$ cannot be at most $8n$ unless $x = y$, hence, at least one of the two is at least $8n + 2$ (being even), and therefore does not fit in the knapsack of size $8n + 1$.

We also note that no two items from the same set S_c can fit in one sack. Let $x, y \in S_c$. Then, for any c' with $c < c'$,

$$x^{(c)}_{c,c',0} + y^{(c)}_{c,c',0} = 2x + 2y$$
$$x^{(c)}_{c,c',1} + y^{(c)}_{c,c',1} = 2(4n - x) + 2(4n - y)$$

Both $2x + 2y$ and $2(4n - x) + 2(4n - y)$ cannot be at most $8n$ unless $x + y = 4n$. [If $2x + 2y \leq 8n$, then, $x + y \leq 4n$, and $2(4n - x) + 2(4n - y) \leq 8n$ implies that $x + y \geq 4n$, or, $x + y = 4n$.] However, x, y are each at most n, and hence $x + y < 2n - 1$. Thus, no two items from the same set S_c can fit in one knapsack.

So, in the case of a common intersection $\{x\}$ of the sets, the d elements $x^{(c)}$, $c = 1, 2, \ldots, t$ fit together in a sack. Assuming all profits of all items to be 1, the total profit is t. In case when the sets are disjoint, then, at most one element from any of the sets may be placed in the sack, with resulting profit 1.

Suppose there is a streaming algorithm for the $t(t-1)$-dimensional knapsack problem with approximation ratio less than $1/t$ and that uses s bits. Player 1 presents the input $\{x^{(1)} : x \in S_1\}$ to this algorithm with all profits set to 1. The state of the algorithm is then sent to player 2, which in turn inserts its input $\{x^{(2)} : x \in S_2\}$ to the state of the algorithm, relays it to player 3 and so on. Finally, player t calls the profit function of the streaming knapsack algorithm. If this profit is 1 or less, it concludes that the sets are disjoint, otherwise, it concludes that the sets have unique common intersection. This protocol correctly determines set disjointness, since, as argued above, for the disjoint case the optimal profit is 1 and is reported as no more than 1 by the streaming knapsack solution. For the unique intersection case, the optimal profit is t and is reported to be greater than 1, since, the approximation ratio is $1/t$. The total communication is $(t-1)$ times the space requirement of the streaming algorithm, namely, s bits. Note that the entries of the vector $x^{(c)}$ are at most $8n$ and at least $epsilon = 1/(2n)$ and the sack size of $8n+1$. Therefore, we can scale the input in such a way that the entries come from $\{\frac{1}{\Delta}, \frac{2}{\Delta}, \ldots, 1\}$ by setting $\Delta = 2n \cdot (8n+1)$.

By the lower bound of the t-party set disjointness problem, $(t-1)s = \Omega(n/t)$ or, $s = \Omega(n/t^2) = \Omega(\sqrt{\Delta}/t^2)$. Thus any streaming algorithm for the $t(t-1)$-dimensional knapsack problem that uses $o(\sqrt{\Delta}/t^2)$ bits must have an approximation ratio at best $1/t$. □

References

1. A. Chakrabarti, S. Khot, and X. Sun. "Near-Optimal Lower Bounds on the Multi-Party Communication Complexity of Set Disjointness". In *Proceedings of International Conference on Computational Complexity (CCC)*, Aarhus, Denmark, 2003.
2. R. Diestel. Graphentheorie. Springer-Verlag, Heidelberg, 2006.
3. Oscar H. Ibarra and Chul E. Kim. "Fast Approximation Algorithms for the Knapsack and the Sum of Subset Problems". *JACM*, 22(4), October 1975.
4. Kazuo Iwama and S. Taketomi. "Removable online knapsack problems". In *Proceedings of International Conference on Automata, Languages and Programming, (ICALP)*, 2002.
5. Kazuo Iwama and Guochuan Zhang. "Optimal Resource Augmentations for Online Knapsack". In *Proceedings of International Workshop on Approximation Algorithms (APPROX)*, pages 180–188, 2007.
6. Richard Karp. "Reducibility among Combinatorial Problems". In *Complexity of Computer Computations, by Miller and Thatcher (eds.) Plenum Press, New York and London*, pages 85–103, 1972.

A Application of Hoeffding's bound in Lemma 7

In this section, we complete the steps in the application of Hoeffding's inequality for the cases used in Lemma 7.

We consider the derivation of (5), (6) and (7).

Case 1. Consider the scenario of (5), where, $|Z_{F,t}| = 1$ and $Z_{F,t} = \{u_t\}$ with $A_{t,u_t} \geq 1/2$. Hoeffding's inequality states that for independent variables $Y_1, \ldots, Y_n$ with each $Y_i \in [0,1]$,

$$\mathsf{Pr}\{Y_1 + Y_2 + \ldots + Y_n > a\} \leq \begin{cases} \exp\{-(a-\mu)^2/(3\mu)\}, & \mu \leq a \leq 2\mu \\ \exp\{-(a-\mu)^2/(a+\mu)\}, & a > 2\mu \end{cases} \quad (10)$$

where, $\mu = \mathsf{E}[Y_1 + \ldots, +Y_n]$.

In the given scenario, $w'_{t,j} = A'_{t,j}/\max_{i \neq u_t} A'_{t,i}$, which lies between 0 and 1. Hence, we can define $Y_j := w'_{t,j} \cdot x_j$, where, x_j is the indicator variable indicating that j has been picked in the sample. . Let $X_t = \sum_{j \neq u_s} w'_{t,j} x_j$. Since, the sampling is done with probability p, it follows that

$$\mathsf{E}[X_t] = \sum_{j \neq u_t} \frac{A'_{t,j} p}{\max_{i \neq u_t} A'_{t,i}} \ .$$

We are interested in the probability of no overflow–which is the event

$$\sum_{j \neq u_t} A'_{t,j} x_j < 1 - A'_{t,u_t} = \beta'_t$$

or, equivalently,

$$X_t < \frac{\beta'_t}{\max_{i \neq u_t} A'_{t,i}} \ .$$

We will use the following combined version of Hoeffding's bound from (10) by slightly relaxing one of the cases.

$$\mathsf{Pr}\{Y_1 + Y_2 + \ldots + Y_n > a\} \leq \exp\{-2(a-\mu)^2/3(a+\mu)\} \ . \quad (11)$$

Using (11), we have

$$\begin{aligned} \mathsf{Pr}\left\{X_t > \beta'_t/(\max_{j \neq u_t} A'_{t,j})\right\} &< \exp\left\{-(2/3)\frac{(\beta'_t - \sum_{j \neq u_t} A'_{t,u_t} p)^2}{(\max_{j \neq u_t} A'_{t,j})(\beta'_t + \sum_{j \neq u_t} A'_{t,u_t} p)}\right\} \\ &\leq \exp\left\{-(2/3)\frac{((1-\epsilon)(1-A_{t,u_t}) - (1+\epsilon)(1-A_{t,u_t})p)^2}{4\epsilon(1-A_{t,u_t})((1-A_{t,u_t}) + (1+\epsilon)(1-A_{t,u_t})p)}\right\} \\ &\quad \{ \text{ since, } \beta'_t \geq (1-\epsilon)(1-A_{t,u_t}) \text{ and} \\ &\quad \text{by Lemma 5, } A_{t,j} \leq 4\epsilon(1-A_{t,u_t})/(1+\epsilon); \text{ so } A'_{t,j} \leq 4\epsilon(1-A_{t,u_t})\} \\ &\leq \exp\left\{-(1/6)\frac{((1-\epsilon) - (1+\epsilon)p)^2}{\epsilon(1+(1+\epsilon)p)}\right\} \ . \end{aligned}$$

This proves the inequality in (5). The inequalities in (6) and (7) are obtained similarly–in these cases, $\max_{j \neq u_t} A'_{t,j} \leq 2\epsilon$ and $\max_{j \neq u_t} A'_{t,j} \leq \epsilon$, respectively.