Set covering with ordered replacement:

Additive and Multiplicative gaps

Laura Sanità

Institute of Mathematics, EPFL, Lausanne, Switzerland

Joint work with:

F. Eisenbrand, N. Kakimura, T. Rothvoß

Aussois, 2011

General Set covering problem

Input:

- A ground set of items $\{1, \ldots, n\}$;
- A set system $\mathcal{S} = \{S_1, \ldots, S_m\}$, with $S_j \subseteq \{1, \ldots, n\}$;

Goal:

- Find min-cardinality $\mathcal{S}' \subseteq \mathcal{S}$, such that $\cup_{S \in \mathcal{S}'} S$ covers all the items.

General Set covering problem

Input:

- A ground set of items $\{1, \ldots, n\}$;
- A set system $\mathcal{S} = \{S_1, \ldots, S_m\}$, with $S_j \subseteq \{1, \ldots, n\}$;

Goal:

- Find min-cardinality $\mathcal{S}' \subseteq \mathcal{S}$, such that $\cup_{S \in \mathcal{S}'} S$ covers all the items.

• Approximation: $O(\log n)$[Chvátal, '79], corresponding hardness result [Feige '98]

Linear Programming Relaxation

- Let $\chi(S) \in \{0,1\}^n$ be the characteristic vector of a set $S \in \mathcal{S}$

$$OPT_f(\mathcal{S}) \quad = \quad \min\Big\{ \sum_{S \in \mathcal{S}} x_S \mid \sum_{S \in \mathcal{S}} x_S \cdot \chi(S) \geq \mathbf{1}, x \geq \mathbf{0} \Big\}$$

Linear Programming Relaxation

- Let $\chi(S) \in \{0,1\}^n$ be the characteristic vector of a set $S \in \mathcal{S}$

$$OPT_f(\mathcal{S}) \quad = \quad \min\Big\{ \sum_{S \in \mathcal{S}} x_S \mid \sum_{S \in \mathcal{S}} x_S \cdot \chi(S) \geq \mathbf{1}, x \geq \mathbf{0} \Big\}$$

- Multiplicative Integrality Gap:

$$\frac{OPT(\mathcal{S})}{OPT_f(\mathcal{S})} = \Theta(\log n)$$

Linear Programming Relaxation

- Let $\chi(S) \in \{0,1\}^n$ be the characteristic vector of a set $S \in \mathcal{S}$

$$OPT_f(\mathcal{S}) \quad = \quad \min\left\{ \sum_{S\in\mathcal{S}} x_S \mid \sum_{S\in\mathcal{S}} x_S \cdot \chi(S) \geq \mathbf{1}, x \geq \mathbf{0} \right\}$$

- Multiplicative Integrality Gap:

$$\frac{OPT(\mathcal{S})}{OPT_f(\mathcal{S})} = \Theta(\log n)$$

- Additive Integrality Gap:

$$OPT(\mathcal{S}) - OPT_f(\mathcal{S}) = \Theta(n)$$

Linear Programming Relaxation

- Let $\chi(S) \in \{0,1\}^n$ be the characteristic vector of a set $S \in \mathcal{S}$

$$OPT_f(\mathcal{S}) \quad = \quad \min\Big\{\sum_{S\in\mathcal{S}} x_S \mid \sum_{S\in\mathcal{S}} x_S \cdot \chi(S) \geq \mathbf{1}, x \geq \mathbf{0}\Big\}$$

- Multiplicative Integrality Gap:

$$\frac{OPT(\mathcal{S})}{OPT_f(\mathcal{S})} = \Theta(\log n)$$

- Additive Integrality Gap:

$$OPT(\mathcal{S}) - OPT_f(\mathcal{S}) = \Theta(n)$$

...but there are some set covering problems more tractable!

Bin Packing

Input:

- Items with sizes $s_1, \ldots, s_n \in [0, 1]$

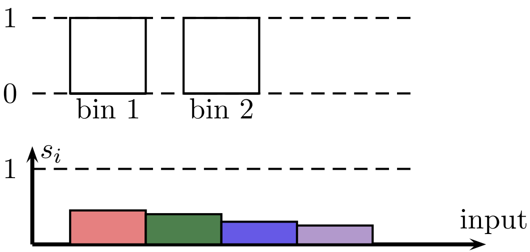
Goal: Pack items into minimum number of **bins** of size 1.

Bin Packing

Input:

- Items with sizes $s_1, \ldots, s_n \in [0, 1]$

Goal: Pack items into minimum number of **bins** of size 1.

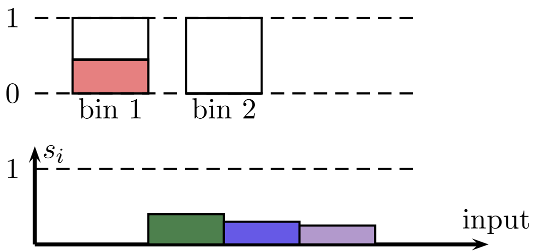


Bin Packing

Input:

- Items with sizes $s_1, \ldots, s_n \in [0, 1]$

Goal: Pack items into minimum number of **bins** of size 1.

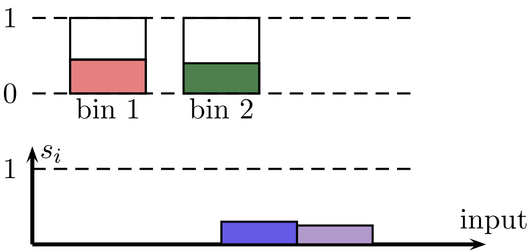


Bin Packing

Input:

- Items with sizes $s_1, \ldots, s_n \in [0, 1]$

Goal: Pack items into minimum number of **bins** of size 1.

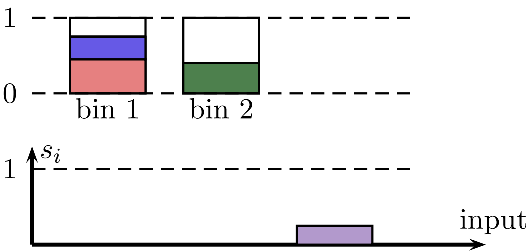


Bin Packing

Input:

- Items with sizes $s_1, \ldots, s_n \in [0, 1]$

Goal: Pack items into minimum number of **bins** of size 1.

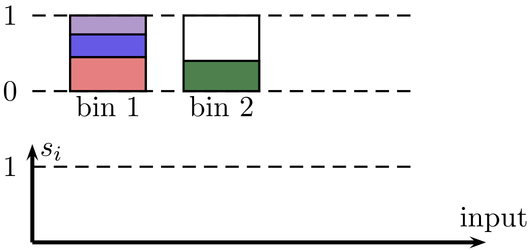


Bin Packing

Input:

- Items with sizes $s_1, \ldots, s_n \in [0, 1]$

Goal: Pack items into minimum number of **bins** of size 1.

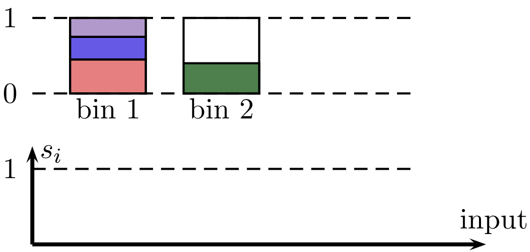


Bin Packing

Input:

- Items with sizes $s_1, \ldots, s_n \in [0, 1]$

Goal: Pack items into minimum number of **bins** of size 1.



- Approximation: Asymptotic FPTAS [Karmarkar & Karp '82]:
$APX \leq OPT + O(\log^2 n)$ in poly-time

Bin Packing: LP relaxation

- Set system $\mathcal{S}$ is given by feasible patterns:

$$\mathcal{S} = \{S \mid \sum_{i \in S} s_i \leq 1\}$$

Bin Packing: LP relaxation

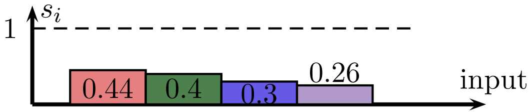
- Set system $\mathcal{S}$ is given by feasible patterns:

$$\mathcal{S} = \{S \mid \sum_{i \in S} s_i \leq 1\}$$

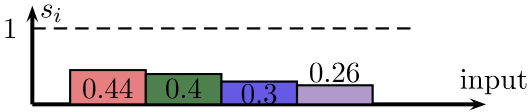
- LP relaxation:

$$\begin{aligned} \min \sum_{S \in \mathcal{S}} x_S & & & \\ \sum_{S \in \mathcal{S}} x_S \cdot \chi(S) & \geq & \mathbf{1} & \\ x_S & \geq & 0 & \forall S \in \mathcal{S} \end{aligned}$$

Bin Packing: LP relaxation - Example



Bin Packing: LP relaxation - Example

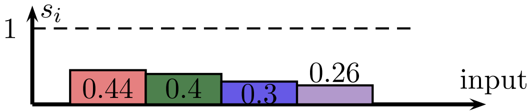


$$\min \sum_{S \in \mathcal{S}} x_S$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} x \geq \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

$$x \geq \mathbf{0}$$

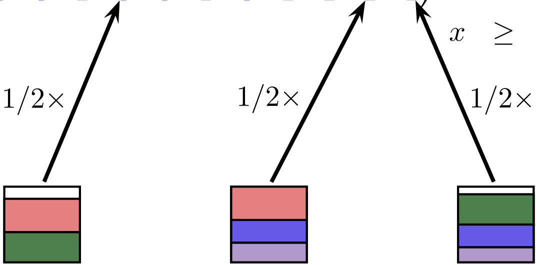
Bin Packing: LP relaxation - Example



$$\min \sum_{S \in \mathcal{S}} x_S$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} x \geq \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

$$x \geq \mathbf{0}$$



Bin Packing: LP relaxation

- LP relaxation:

$$\begin{aligned} \min \sum_{S \in \mathcal{S}} x_S & \\ \sum_{S \in \mathcal{S}} x_S \cdot \chi(S) &\geq \mathbf{1} \\ x_S &\geq 0 \quad \forall S \in \mathcal{S} \end{aligned}$$

Bin Packing: LP relaxation

- LP relaxation:

$$\begin{aligned} \min \sum_{S \in \mathcal{S}} x_S & & \\ \sum_{S \in \mathcal{S}} x_S \cdot \chi(S) &\geq \mathbf{1} & \\ x_S &\geq 0 & \forall S \in \mathcal{S} \end{aligned}$$

- Additive Integrality gap: $O(\log^2 n)$ [Karmarkar & Karp '82]

Bin Packing: LP relaxation

- LP relaxation:

$$\min \sum_{S \in \mathcal{S}} x_S$$
$$\sum_{S \in \mathcal{S}} x_S \cdot \chi(S) \geq \mathbf{1}$$
$$x_S \geq 0 \quad \forall S \in \mathcal{S}$$

- Additive Integrality gap: $O(\log^2 n)$ [Karmarkar & Karp '82]

Modified Integer Roundup Conjecture:

$OPT(\mathcal{S}) \leq \lceil OPT_f(\mathcal{S}) \rceil + 1$

Bin Packing: LP relaxation

- LP relaxation:

$$\min \sum_{S \in \mathcal{S}} x_S$$

$$\sum_{S \in \mathcal{S}} x_S \cdot \chi(S) \geq \mathbf{1}$$

$$x_S \geq 0 \quad \forall S \in \mathcal{S}$$

- Additive Integrality gap: $O(\log^2 n)$ [Karmarkar & Karp '82]

Modified Integer Roundup Conjecture:

$OPT(\mathcal{S}) \leq \lceil OPT_f(\mathcal{S}) \rceil + 1$

- Multiplicative Integrality gap: $O(1)$

Question: which properties make Bin packing so special?

Question: which properties make Bin packing so special?

...Possible answer: the **ordered replacement**!

Set cover with ordered replacement

Input:

- A ground set of items $\{1, \ldots, n\}$
- A set system $\mathcal{S}$ defined on the ground set

Goal:

- Find min-cardinality $\mathcal{S}' \subseteq \mathcal{S}$, such that $\cup_{S \in \mathcal{S}'} S$ covers all the items.

Set cover with ordered replacement

Input:

- A ground set of items $\{1, \ldots, n\}$ with a total order $\preceq$
- A set system $\mathcal{S}$ defined on the ground set

Goal:

- Find min-cardinality $\mathcal{S}' \subseteq \mathcal{S}$, such that $\cup_{S \in \mathcal{S}'} S$ covers all the items.

Set cover with ordered replacement

Input:

- A ground set of items $\{1, \ldots, n\}$ with a total order $\preceq$
- A set system $\mathcal{S}$ defined on the ground set respecting the given order:

$$S \in \mathcal{S}, i \in S, j \notin S, j \preceq i \Rightarrow ((S \backslash \{i\}) \cup \{j\}) \in \mathcal{S}$$

Goal:

- Find min-cardinality $\mathcal{S}' \subseteq \mathcal{S}$, such that $\cup_{S \in \mathcal{S}'} S$ covers all the items.

Set cover with ordered replacement

Input:

- A ground set of items $\{1, \ldots, n\}$ with a total order $\preceq$
- A set system $\mathcal{S}$ defined on the ground set respecting the given order:

$$S \in \mathcal{S}, i \in S, j \notin S, j \preceq i \Rightarrow ((S \backslash \{i\}) \cup \{j\}) \in \mathcal{S}$$

Goal:

- Find min-cardinality $\mathcal{S}' \subseteq \mathcal{S}$, such that $\cup_{S \in \mathcal{S}'} S$ covers all the items.

• Examples: Cardinality Bin packing, Open end Bin packing, Bin packing with general cost structure, some scheduling problems and ordered vector packing problems ...

Set cover with ordered replacement

Input:

- A ground set of items $\{1, \ldots, n\}$ with a total order $\preceq$
- A set system $\mathcal{S}$ defined on the ground set respecting the given order:

$$S \in \mathcal{S}, i \in S, j \notin S, j \preceq i \Rightarrow ((S \backslash \{i\}) \cup \{j\}) \in \mathcal{S}$$

Goal:

- Find min-cardinality $\mathcal{S}' \subseteq \mathcal{S}$, such that $\cup_{S \in \mathcal{S}'} S$ covers all the items.

• Examples: Cardinality Bin packing, Open end Bin packing, Bin packing with general cost structure, some scheduling problems and ordered vector packing problems ...

What can we say in this general setting?

Our Results

Our Results

Theorem

For the Set cover with ordered replacement problem, the additive integrality gap is $O(\log^3 n)$ *and* $\Omega(\log n)$.

Our Results

Theorem

For the Set cover with ordered replacement problem, the additive integrality gap is $O(\log^3 n)$ and $\Omega(\log n)$.

Theorem

Unless $P = NP$, Set cover with ordered replacement does not allow an asymptotic polynomial time approximation scheme (APTAS).

Our Results

Theorem

For the Set cover with ordered replacement problem, the additive integrality gap is $O(\log^3 n)$ *and* $\Omega(\log n)$.

Theorem

Unless $P = NP$, *Set cover with ordered replacement does not allow an asymptotic polynomial time approximation scheme (APTAS).*

Theorem

For the Set cover with ordered replacement problem, the multiplicative integrality gap is $\Theta(\log \log n)$.

Additive Integrality gap

Theorem

Given an instance $\mathcal{S}$ of Set cover with ordered replacement, the additive integrality gap is $O(\log^3 n)$.

Additive Integrality gap

Theorem

Given an instance $\mathcal{S}$ of Set cover with ordered replacement, the additive integrality gap is $O(\log^3 n)$.

- As in [Karmarkar & Karp '82], we will construct an integer solution from a fractional one by doing a sequence of iterations.

Additive Integrality gap

Theorem

Given an instance $\mathcal{S}$ of Set cover with ordered replacement, the additive integrality gap is $O(\log^3 n)$.

- As in [Karmarkar & Karp '82], we will construct an integer solution from a fractional one by doing a sequence of iterations.
- At each iteration:
  - cover part of our elements by rounding down a fractional solution;
  - modify the residual instance;
  - re-optimize.

Preliminaries of the proof

- We will consider the following more general LP:

$$\min \sum_{S \in \mathcal{S}} x_S$$

$$\begin{aligned} \sum_{S \in \mathcal{S}} x_S \cdot \chi(S) &\geq \mathbf{b} \\ x_S &\geq 0 \quad \forall S \in \mathcal{S} \end{aligned}$$

Preliminaries of the proof

- We will consider the following more general LP:

$$\begin{aligned} \min \sum_{S \in \mathcal{S}} x_S & \\ \sum_{S \in \mathcal{S}} x_S \cdot \chi(S) &\geq \mathbf{b} \\ x_S &\geq 0 \quad \forall S \in \mathcal{S} \end{aligned}$$

- $\mathbf{b}$ is the vector of multiplicity for the items:

Preliminaries of the proof

- We will consider the following more general LP:

$$\begin{aligned} \min \sum_{S \in \mathcal{S}} x_S & & \\ \sum_{S \in \mathcal{S}} x_S \cdot \chi(S) & \geq & \mathbf{b} \\ x_S & \geq & 0 \quad \forall S \in \mathcal{S} \end{aligned}$$

- $\mathbf{b}$ is the vector of multiplicity for the items:

if e.g. $i \preceq j$ and $j \preceq i$, “delete” i and keep j with multiplicity 2.

Preliminaries of the proof

- We will consider the following more general LP:

$$\min \sum_{S \in \mathcal{S}} x_S$$
$$\sum_{S \in \mathcal{S}} x_S \cdot \chi(S) \geq \mathbf{b}$$
$$x_S \geq 0 \quad \forall S \in \mathcal{S}$$

- $\mathbf{b}$ is the vector of multiplicity for the items:

if e.g. $i \preceq j$ and $j \preceq i$, "delete" i and keep j with multiplicity 2.

- We will reduce # of constraints at the expense of increasing multiplicity, applying a grouping technique similar to [K & K '82].

Preliminaries of the proof

- We will consider the following more general LP:

$$\min \sum_{S \in \mathcal{S}} x_S$$
$$\sum_{S \in \mathcal{S}} x_S \cdot \chi(S) \geq \mathbf{b}$$
$$x_S \geq 0 \quad \forall S \in \mathcal{S}$$

- **b** is the vector of multiplicity for the items:

if e.g. $i \preceq j$ and $j \preceq i$, "delete" i and keep j with multiplicity 2.

- We will reduce # of constraints at the expense of increasing multiplicity, applying a grouping technique similar to [K & K '82].

- ...but the grouping of [K & K '82] crucially relies on the given item sizes of the Bin packing instace, which are missing here...!!

Idea: Pseudo-sizes

Idea: Pseudo-sizes

- We will define pseudo-sizes $s_i \in]0,1]$ $\forall$ item i, satisfying:
  - $s_j \leq s_i$ if $j \preceq i$; (i)
  - we can cover any subset I of items with $O(\sum_{i \in I} s_i) + O(\log \frac{1}{s_{min}})$ sets. (ii)

Idea: Pseudo-sizes

- We will define pseudo-sizes $s_i \in]0,1]$ $\forall$ item i, satisfying:
  - $s_j \le s_i$ if $j \preceq i$; (i)
  - we can cover any subset I of items with $O(\sum_{i \in I} s_i) + O(\log \frac{1}{s_{min}})$ sets. (ii)

Lemma

Let $\mathcal{S}$ be an instance of Set cover with ordered replacement. Let s be a vector of pseudo-sizes satisfying (i) and (ii), and $\alpha := \max_{S \in \mathcal{S}} \sum_{i \in S} s_i$. Then $OPT(\mathcal{S}, b) - OPT_f(\mathcal{S}, b) = O(\alpha \cdot \log(\frac{1}{s_{min}}) \cdot \log n)$.

Idea: Pseudo-sizes

- We will define pseudo-sizes $s_i \in]0,1]$ $\forall$ item i, satisfying:
  - $s_j \leq s_i$ if $j \preceq i$; (i)
  - we can cover any subset I of items with $O(\sum_{i \in I} s_i) + O(\log \frac{1}{s_{min}})$ sets. (ii)

Lemma

Let $\mathcal{S}$ be an instance of Set cover with ordered replacement. Let s be a vector of pseudo-sizes satisfying (i) and (ii), and $\alpha := \max_{S \in \mathcal{S}} \sum_{i \in S} s_i$.
Then $OPT(\mathcal{S}, b) - OPT_f(\mathcal{S}, b) = O(\alpha \cdot \log(\frac{1}{s_{min}}) \cdot \log n)$.

- For Bin packing, the original item sizes satisfy (i) and (ii).

Idea: Pseudo-sizes

- We will define pseudo-sizes $s_i \in]0,1]$ $\forall$ item i, satisfying:
  - $s_j \leq s_i$ if $j \preceq i$; (i)
  - we can cover any subset I of items with $O(\sum_{i \in I} s_i) + O(\log \frac{1}{s_{min}})$ sets. (ii)

Lemma

Let $\mathcal{S}$ be an instance of Set cover with ordered replacement. Let s be a vector of pseudo-sizes satisfying (i) and (ii), and $\alpha := \max_{S \in \mathcal{S}} \sum_{i \in S} s_i$.
Then $OPT(\mathcal{S}, b) - OPT_f(\mathcal{S}, b) = O(\alpha \cdot \log(\frac{1}{s_{min}}) \cdot \log n)$.

- For Bin packing, the original item sizes satisfy (i) and (ii).

Still, $\alpha \leq 1$ and w.l.o.g. $s_{min} \geq 1/n$.

Idea: Pseudo-sizes

- We will define pseudo-sizes $s_i \in]0,1]$ $\forall$ item i, satisfying:
  - $s_j \leq s_i$ if $j \preceq i$; (i)
  - we can cover any subset I of items with $O(\sum_{i \in I} s_i) + O(\log \frac{1}{s_{min}})$ sets. (ii)

Lemma

Let $\mathcal{S}$ be an instance of Set cover with ordered replacement. Let s be a vector of pseudo-sizes satisfying (i) and (ii), and $\alpha := \max_{S \in \mathcal{S}} \sum_{i \in S} s_i$.
Then $OPT(\mathcal{S}, b) - OPT_f(\mathcal{S}, b) = O(\alpha \cdot \log(\frac{1}{s_{min}}) \cdot \log n)$.

- For Bin packing, the original item sizes satisfy (i) and (ii).

Still, $\alpha \leq 1$ and w.l.o.g. $s_{min} \geq 1/n$. That is, our Lemma yelds the result of [Karmarkar & Karp '82].

Proof of the Lemma: sketch – 1/2

At each iteration:

Proof of the Lemma: sketch – 1/2

At each iteration:

- Consider an opt vertex solution x^* of the LP and buy $\lfloor x^* \rfloor$

Proof of the Lemma: sketch – 1/2

At each iteration:

- Consider an opt vertex solution x^* of the LP and buy $\lfloor x^* \rfloor$
- Define a residual instance by $b'_i := \sum_{S: i \in S} (x^*_S - \lfloor x^*_S \rfloor)$.

Proof of the Lemma: sketch – 1/2

At each iteration:

- Consider an opt vertex solution x^* of the LP and buy $\lfloor x^* \rfloor$
- Define a residual instance by $b'_i := \sum_{S:i\in S}(x^*_S - \lfloor x^*_S \rfloor)$.

Obs 1: $OPT(\mathcal{S}, b) - OPT_f(\mathcal{S}, b) \leq OPT(\mathcal{S}, b') - OPT_f(\mathcal{S}, b')$

Proof of the Lemma: sketch – 1/2

At each iteration:

- Consider an opt vertex solution x^* of the LP and buy $\lfloor x^* \rfloor$
- Define a residual instance by $b'_i := \sum_{S:i \in S}(x^*_S - \lfloor x^*_S \rfloor)$.

Obs 1: $OPT(\mathcal{S}, b) - OPT_f(\mathcal{S}, b) \leq OPT(\mathcal{S}, b') - OPT_f(\mathcal{S}, b')$

Obs 2: $\sum_i s_i \cdot b'_i = \sum_i s_i \sum_{S:i \in S}(x^*_S - \lfloor x^*_S \rfloor) \leq \alpha \text{ support}(b)$

Proof of the Lemma: sketch – 1/2

At each iteration:

- Consider an opt vertex solution x^* of the LP and buy $\lfloor x^* \rfloor$
- Define a residual instance by $b'_i := \sum_{S: i \in S} (x^*_S - \lfloor x^*_S \rfloor)$.

Obs 1: $OPT(\mathcal{S}, b) - OPT_f(\mathcal{S}, b) \leq OPT(\mathcal{S}, b') - OPT_f(\mathcal{S}, b')$

Obs 2: $\sum_i s_i \cdot b'_i = \sum_i s_i \sum_{S: i \in S} (x^*_S - \lfloor x^*_S \rfloor) \leq \alpha \text{ support}(b)$

- Partition the items into *classes*:

$$U_\ell = \left\{ i \mid \left(\frac{1}{2}\right)^{\ell+1} < s_i \leq \left(\frac{1}{2}\right)^{\ell} \right\} \text{ for } \ell = 0, \ldots, \lfloor \log(1/s_{\min}) \rfloor.$$

Proof of the Lemma: sketch – 1/2

At each iteration:

- Consider an opt vertex solution x^* of the LP and buy $\lfloor x^* \rfloor$
- Define a residual instance by $b'_i := \sum_{S:i\in S}(x^*_S - \lfloor x^*_S \rfloor)$.

Obs 1: $OPT(\mathcal{S}, b) - OPT_f(\mathcal{S}, b) \leq OPT(\mathcal{S}, b') - OPT_f(\mathcal{S}, b')$

Obs 2: $\sum_i s_i \cdot b'_i = \sum_i s_i \sum_{S:i\in S}(x^*_S - \lfloor x^*_S \rfloor) \leq \alpha \text{ support}(b)$

- Partition the items into *classes*:

$$U_\ell = \left\{ i \mid \left(\frac{1}{2}\right)^{\ell+1} < s_i \leq \left(\frac{1}{2}\right)^{\ell} \right\} \text{ for } \ell = 0, \ldots, \lfloor \log(1/s_{\min}) \rfloor.$$

- For each such class U_ℓ, build groups of $4 \cdot 2^\ell \alpha$ consecutive elements, and discard the first and the last group.

Proof of the Lemma: sketch – 1/2

At each iteration:

- Consider an opt vertex solution x^* of the LP and buy $\lfloor x^* \rfloor$
- Define a residual instance by $b'_i := \sum_{S:i \in S}(x^*_S - \lfloor x^*_S \rfloor)$.

Obs 1: $OPT(\mathcal{S}, b) - OPT_f(\mathcal{S}, b) \leq OPT(\mathcal{S}, b') - OPT_f(\mathcal{S}, b')$

Obs 2: $\sum_i s_i \cdot b'_i = \sum_i s_i \sum_{S:i \in S}(x^*_S - \lfloor x^*_S \rfloor) \leq \alpha \text{ support}(b)$

- Partition the items into *classes*:

$$U_\ell = \left\{ i \mid \left(\frac{1}{2}\right)^{\ell+1} < s_i \leq \left(\frac{1}{2}\right)^{\ell} \right\} \text{ for } \ell = 0, \ldots, \lfloor \log(1/s_{\min}) \rfloor.$$

- For each such class U_ℓ, build groups of $4 \cdot 2^\ell \alpha$ consecutive elements, and discard the first and the last group.

- Total size of discarded elements $\leq 8 \cdot \alpha \cdot (\log(1/s_{\min}) + 1)$. By *(ii)*, we can cover them with $O(\alpha \cdot \log(1/s_{\min}))$ sets.

Proof of the Lemma: sketch – 2/2

- We "round-up" the elements in each group to the largest one: i.e. we have one element type with multiplicity $4 \cdot 2^{\ell}\alpha$

Proof of the Lemma: sketch – 2/2

- We "round-up" the elements in each group to the largest one: i.e. we have one element type with multiplicity $4 \cdot 2^{\ell}\alpha$
- This way, we have a new "rounded" instance encoded by b''

Proof of the Lemma: sketch – 2/2

- We “round-up” the elements in each group to the largest one: i.e. we have one element type with multiplicity $4 \cdot 2^{\ell}\alpha$
- This way, we have a new “rounded” instance encoded by b''

Obs 1: $OPT_f(\mathcal{S}, b'') \leq OPT_f(\mathcal{S}, b')$
(discarded groups compensate the round-up operation!)

Obs 2: $OPT(\mathcal{S}, b') \leq OPT(\mathcal{S}, b'') + O(\alpha \cdot \log(1/s_{\min}))$

Proof of the Lemma: sketch – 2/2

- We "round-up" the elements in each group to the largest one: i.e. we have one element type with multiplicity $4 \cdot 2^{\ell}\alpha$
- This way, we have a new "rounded" instance encoded by b''

Obs 1: $OPT_f(\mathcal{S}, b'') \leq OPT_f(\mathcal{S}, b')$
(discarded groups compensate the round-up operation!)

Obs 2: $OPT(\mathcal{S}, b') \leq OPT(\mathcal{S}, b'') + O(\alpha \cdot \log(1/s_{\min}))$

- $OPT(\mathcal{S}, b') - OPT_f(\mathcal{S}, b') \leq OPT(\mathcal{S}, b'') - OPT_f(\mathcal{S}, b'') + O(\alpha \cdot \log(1/s_{\min}))$

Proof of the Lemma: sketch – 2/2

- We "round-up" the elements in each group to the largest one: i.e. we have one element type with multiplicity $4 \cdot 2^{\ell}\alpha$
- This way, we have a new "rounded" instance encoded by b''

Obs 1: $OPT_f(\mathcal{S}, b'') \leq OPT_f(\mathcal{S}, b')$
(discarded groups compensate the round-up operation!)

Obs 2: $OPT(\mathcal{S}, b') \leq OPT(\mathcal{S}, b'') + O(\alpha \cdot \log(1/s_{\min}))$

- $OPT(\mathcal{S}, b') - OPT_f(\mathcal{S}, b') \leq OPT(\mathcal{S}, b'') - OPT_f(\mathcal{S}, b'') + O(\alpha \cdot \log(1/s_{\min}))$
- Support of b'' is the number of non-discarded groups

Proof of the Lemma: sketch – 2/2

- We "round-up" the elements in each group to the largest one: i.e. we have one element type with multiplicity $4 \cdot 2^\ell \alpha$
- This way, we have a new "rounded" instance encoded by b''

Obs 1: $OPT_f(\mathcal{S}, b'') \leq OPT_f(\mathcal{S}, b')$
(discarded groups compensate the round-up operation!)

Obs 2: $OPT(\mathcal{S}, b') \leq OPT(\mathcal{S}, b'') + O(\alpha \cdot \log(1/s_{\min}))$

- $OPT(\mathcal{S}, b') - OPT_f(\mathcal{S}, b') \leq OPT(\mathcal{S}, b'') - OPT_f(\mathcal{S}, b'') + O(\alpha \cdot \log(1/s_{\min}))$
- Support of b'' is the number of non-discarded groups
- Each group contains $4 \cdot 2^\ell \alpha$ items of sizes $\geq \frac{1}{2^{\ell+1}}$

Proof of the Lemma: sketch – 2/2

- We "round-up" the elements in each group to the largest one: i.e. we have one element type with multiplicity $4 \cdot 2^\ell \alpha$
- This way, we have a new "rounded" instance encoded by b''

Obs 1: $OPT_f(\mathcal{S}, b'') \leq OPT_f(\mathcal{S}, b')$
(discarded groups compensate the round-up operation!)

Obs 2: $OPT(\mathcal{S}, b') \leq OPT(\mathcal{S}, b'') + O(\alpha \cdot \log(1/s_{\min}))$

- $OPT(\mathcal{S}, b') - OPT_f(\mathcal{S}, b') \leq OPT(\mathcal{S}, b'') - OPT_f(\mathcal{S}, b'') + O(\alpha \cdot \log(1/s_{\min}))$
- Support of b'' is the number of non-discarded groups
- Each group contains $4 \cdot 2^\ell \alpha$ items of sizes $\geq \frac{1}{2^{\ell+1}}$
$\Rightarrow$ Total size of the group $\geq 2\alpha$

Proof of the Lemma: sketch – 2/2

- We "round-up" the elements in each group to the largest one: i.e. we have one element type with multiplicity $4 \cdot 2^{\ell}\alpha$
- This way, we have a new "rounded" instance encoded by b''

Obs 1: $OPT_f(\mathcal{S}, b'') \leq OPT_f(\mathcal{S}, b')$
(discarded groups compensate the round-up operation!)

Obs 2: $OPT(\mathcal{S}, b') \leq OPT(\mathcal{S}, b'') + O(\alpha \cdot \log(1/s_{\min}))$

- $OPT(\mathcal{S}, b') - OPT_f(\mathcal{S}, b') \leq OPT(\mathcal{S}, b'') - OPT_f(\mathcal{S}, b'') + O(\alpha \cdot \log(1/s_{\min}))$
- Support of b'' is the number of non-discarded groups
- Each group contains $4 \cdot 2^{\ell}\alpha$ items of sizes $\geq \frac{1}{2^{\ell+1}}$
$\Rightarrow$ Total size of the group $\geq 2\alpha$
$\Rightarrow 2\alpha$ support$(b'') \leq \sum_i s_i \cdot b'_i$

Proof of the Lemma: sketch – 2/2

- We "round-up" the elements in each group to the largest one: i.e. we have one element type with multiplicity $4 \cdot 2^{\ell}\alpha$
- This way, we have a new "rounded" instance encoded by b''

Obs 1: $OPT_f(\mathcal{S}, b'') \leq OPT_f(\mathcal{S}, b')$
(discarded groups compensate the round-up operation!)

Obs 2: $OPT(\mathcal{S}, b') \leq OPT(\mathcal{S}, b'') + O(\alpha \cdot \log(1/s_{\min}))$

- $OPT(\mathcal{S}, b') - OPT_f(\mathcal{S}, b') \leq OPT(\mathcal{S}, b'') - OPT_f(\mathcal{S}, b'') + O(\alpha \cdot \log(1/s_{\min}))$
- Support of b'' is the number of non-discarded groups
- Each group contains $4 \cdot 2^{\ell}\alpha$ items of sizes $\geq \frac{1}{2^{\ell+1}}$
$\Rightarrow$ Total size of the group $\geq 2\alpha$
$\Rightarrow 2\alpha$ support$(b'') \leq \sum_i s_i \cdot b'_i \leq \alpha$ support(b).
- support$(b'') \leq$ support$(b)/2$

Proof of the Theorem

Proof of the Theorem

- To prove the theorem, it remains to define good pseudo-sizes for any set system $\mathcal{S}$.

Proof of the Theorem

- To prove the theorem, it remains to define good pseudo-sizes for any set system $\mathcal{S}$.

We let:

$$s_i := \min\left\{\frac{1}{|S|} \mid S \in \mathcal{S} \text{ contains only elements } j \succeq i\right\}$$

Proof of the Theorem

- To prove the theorem, it remains to define good pseudo-sizes for any set system $\mathcal{S}$.

We let:

$$s_i := \min\left\{\frac{1}{|S|} \mid S \in \mathcal{S} \text{ contains only elements } j \succeq i\right\}$$

- Condition (i) : $s_j \leq s_i$ if $j \preceq i$;

Proof of the Theorem

- To prove the theorem, it remains to define good pseudo-sizes for any set system $\mathcal{S}$.

We let:

$$s_i := \min\left\{\frac{1}{|S|} \mid S \in \mathcal{S} \text{ contains only elements } j \succeq i\right\}$$

- Condition (i) : $s_j \leq s_i$ if $j \preceq i$;
- Condition (ii): Cover items in I with $O(\sum_{i \in I} s_i) + O(\log \frac{1}{s_{min}})$ sets

Proof of the Theorem

- To prove the theorem, it remains to define good pseudo-sizes for any set system $\mathcal{S}$.

We let:

$$s_i := \min\left\{\frac{1}{|S|} \mid S \in \mathcal{S} \text{ contains only elements } j \succeq i\right\}$$

- Condition (i) : $s_j \leq s_i$ if $j \preceq i$;
- Condition (ii): Cover items in I with $O(\sum_{i \in I} s_i) + O(\log \frac{1}{s_{min}})$ sets (partition again the elements into classes according to the sizes, and iteratively cover the largest uncovered elements...)

Proof of the Theorem

- To prove the theorem, it remains to define good pseudo-sizes for any set system $\mathcal{S}$.

We let:

$$s_i := \min\left\{\frac{1}{|S|} \mid S \in \mathcal{S} \text{ contains only elements } j \succeq i\right\}$$

- Condition (i) : $s_j \leq s_i$ if $j \preceq i$;
- Condition (ii): Cover items in I with $O(\sum_{i \in I} s_i) + O(\log \frac{1}{s_{min}})$ sets (partition again the elements into classes according to the sizes, and iteratively cover the largest uncovered elements...)
- Note: For every S, $\sum_{i \in S} s_i \leq \sum_{i=1}^{n} \frac{1}{n}$, $\Rightarrow \alpha = O(\log n)$

Proof of the Theorem

- To prove the theorem, it remains to define good pseudo-sizes for any set system $\mathcal{S}$.

We let:

$$s_i := \min\left\{\frac{1}{|S|} \mid S \in \mathcal{S} \text{ contains only elements } j \succeq i\right\}$$

- Condition (i) : $s_j \leq s_i$ if $j \preceq i$;
- Condition (ii): Cover items in I with $O(\sum_{i \in I} s_i) + O(\log \frac{1}{s_{min}})$ sets (partition again the elements into classes according to the sizes, and iteratively cover the largest uncovered elements...)
- Note: For every S, $\sum_{i \in S} s_i \leq \sum_{i=1}^{n} \frac{1}{n}$, $\Rightarrow \alpha = O(\log n)$
- By the Lemma,

$$\text{additive gap is } O(\alpha \cdot \log(\tfrac{1}{s_{min}}) \cdot \log n) = O(\log^3 n).$$

Final Remarks

Final Remarks

- We can extend bound on the additive integrality gap, in the case where the replacement order is not a total order.
The bound is $O(d^2 \log^3 n)$, where d is the *Dilworth number* of the partial order.

Final Remarks

- We can extend bound on the additive integrality gap, in the case where the replacement order is not a total order.
The bound is $O(d^2 \log^3 n)$, where d is the *Dilworth number* of the partial order.

- All the given bounds hold in case of weighted sets.

Final Remarks

- We can extend bound on the additive integrality gap, in the case where the replacement order is not a total order.
The bound is $O(d^2 \log^3 n)$, where d is the *Dilworth number* of the partial order.

- All the given bounds hold in case of weighted sets.

- Open questions:
  - Is there a $O(1)$-apx for Set cover with ordered replacement? (we can prove a 2-apx in quasi-polynomial time)
  - Can we give a bound on the additive integrality gap better than $O(\log^3 n)$?
  - Is the additive integrality gap constant for Bin packing?