The Entropy Rounding Method in Approximation Algorithms

Thomas Rothvoß

Department of Mathematics, M.I.T.

Cargèse 2011

Massachusetts Institute of Technology

Alexander von Humboldt Stiftung/Foundation

A general rounding problem

Problem:

- **Given:** $A \in \mathbb{R}^{n \times m}$, fractional solution $x \in [0,1]^m$
- **Find:** $y \in \{0,1\}^m : Ax \approx Ay$

A general rounding problem

Problem:

- **Given:** $A \in \mathbb{R}^{n \times m}$, fractional solution $x \in [0,1]^m$
- **Find:** $y \in \{0,1\}^m : Ax \approx Ay$

General strategies:

- **Randomized rounding:**
  Flip $\Pr[y_i = 1] = x_i$ (in)dependently

A general rounding problem

Problem:

- **Given:** $A \in \mathbb{R}^{n\times m}$, fractional solution $x \in [0,1]^m$
- **Find:** $y \in \{0,1\}^m : Ax \approx Ay$

General strategies:

- **Randomized rounding:**
  Flip $\Pr[y_i = 1] = x_i$ (in)dependently
- **Use properties of basic solutions:**
  $|\mathrm{supp}(x)| \leq \#\mathrm{rows}(A)$

A general rounding problem

Problem:

- **Given:** $A \in \mathbb{R}^{n \times m}$, fractional solution $x \in [0,1]^m$
- **Find:** $y \in \{0,1\}^m : Ax \approx Ay$

General strategies:

- **Randomized rounding:**
  Flip $\Pr[y_i = 1] = x_i$ (in)dependently
- **Use properties of basic solutions:**
  $|\mathrm{supp}(x)| \leq \#\mathrm{rows}(A)$

Try another way:

- "Entropy rounding method" based on discrepancy theory

A general rounding problem

Problem:

- **Given:** $A \in \mathbb{R}^{n \times m}$, fractional solution $x \in [0,1]^m$
- **Find:** $y \in \{0,1\}^m : Ax \approx Ay$

General strategies:

- **Randomized rounding:**
  Flip $\Pr[y_i = 1] = x_i$ (in)dependently
- **Use properties of basic solutions:**
  $|\mathrm{supp}(x)| \leq \#\mathrm{rows}(A)$

Try another way:

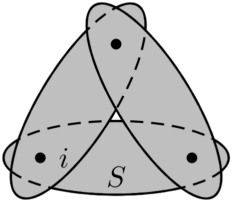
- "Entropy rounding method" based on discrepancy theory

Application:

- A $(OPT + O(\log^2 OPT))$-algorithm for **Bin Packing With Rejection**

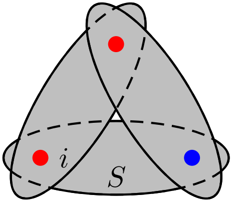
Discrepancy theory

- Set system $\mathcal{S} = \{S_1, \ldots, S_m\}, S_i \subseteq [n]$

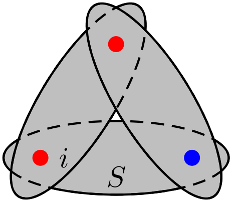


Discrepancy theory

- Set system $\mathcal{S} = \{S_1, \ldots, S_m\}, S_i \subseteq [n]$
- **Coloring** $\chi : [n] \to \{-1, +1\}$



Discrepancy theory

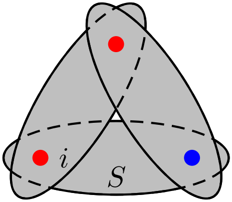


- Set system $\mathcal{S} = \{S_1, \dots, S_m\}, S_i \subseteq [n]$
- **Coloring** $\chi : [n] \to \{-1, +1\}$
- **Discrepancy**

$$\mathrm{disc}(\mathcal{S}) = \min_{\chi:[n]\to\{\pm 1\}} \max_{S \in \mathcal{S}} |\chi(S)|.$$

where $\chi(S) = \sum_{i \in S} \chi(i)$.

Discrepancy theory



- Set system $\mathcal{S} = \{S_1, \ldots, S_m\}, S_i \subseteq [n]$
- **Coloring** $\chi : [n] \to \{-1, +1\}$
- **Discrepancy**

$$\mathrm{disc}(\mathcal{S}) = \min_{\chi:[n]\to\{\pm 1\}} \max_{S\in\mathcal{S}} |\chi(S)|.$$

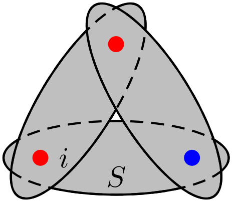
where $\chi(S) = \sum_{i\in S} \chi(i)$.

Known results:

- n sets, n elements: $\mathrm{disc}(\mathcal{S}) = O(\sqrt{n})$ [Spencer '85]

Discrepancy theory

- Set system $\mathcal{S} = \{S_1, \ldots, S_m\}, S_i \subseteq [n]$
- **Coloring** $\chi : [n] \to \{-1, +1\}$
- **Discrepancy**



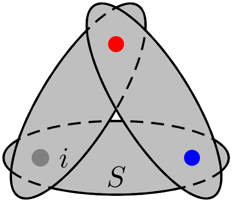
$$\mathrm{disc}(\mathcal{S}) = \min_{\chi:[n]\to\{\pm1\}} \max_{S\in\mathcal{S}} |\chi(S)|.$$

where $\chi(S) = \sum_{i \in S} \chi(i)$.

Known results:

- n sets, n elements: $\mathrm{disc}(\mathcal{S}) = O(\sqrt{n})$ [Spencer '85]
- Every element in $\leq t$ sets: $\mathrm{disc}(\mathcal{S}) < 2t$ [Beck & Fiala '81]
  Conjecture: $\mathrm{disc}(\mathcal{S}) \leq O(\sqrt{t})$

Discrepancy theory



- Set system $\mathcal{S} = \{S_1, \ldots, S_m\}, S_i \subseteq [n]$
- **Coloring** $\chi : [n] \to \{-1, +1\}$
- **Discrepancy**

$$\mathrm{disc}(\mathcal{S}) = \min_{\chi:[n]\to\{\pm 1\}} \max_{S\in\mathcal{S}} |\chi(S)|.$$

where $\chi(S) = \sum_{i\in S} \chi(i)$.

Known results:

- n sets, n elements: $\mathrm{disc}(\mathcal{S}) = O(\sqrt{n})$ [Spencer '85]
- Every element in $\leq t$ sets: $\mathrm{disc}(\mathcal{S}) < 2t$ [Beck & Fiala '81]
  Conjecture: $\mathrm{disc}(\mathcal{S}) \leq O(\sqrt{t})$

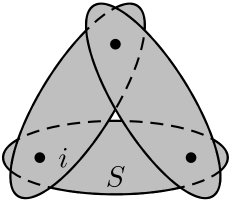
More definitions:

- **Partial coloring:** $\chi : [n] \to \{0, -1, +1\}$
- **Half coloring:** $\chi : [n] \to \{0, -1, +1\}$, $|\mathrm{supp}(\chi)| \geq n/2$

Matrix discrepancy

- Matrix A

$$\mathrm{disc}(A) = \min_{\chi \in \{\pm 1\}^n} \|A\chi\|_\infty$$



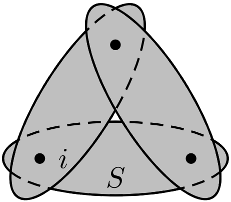
$$A = \begin{array}{cl} \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix} & \begin{array}{l} \\ \text{set } S \\ \\ \end{array} \\ \begin{array}{ccc} & i & \end{array} & \end{array}$$

$$\mathrm{disc}(\mathcal{S}) \quad = \quad \mathrm{disc}(A)$$

Matrix discrepancy

- Matrix A

$$\mathrm{disc}(A) = \min_{\chi \in \{\pm 1\}^n} \|A\chi\|_\infty$$



$$A = \begin{array}{cl} \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix} & \begin{array}{l} \\ \text{set } S \\ \\ \end{array} \\ \begin{array}{ccc} & i & \end{array} & \end{array}$$

$$\mathrm{disc}(\mathcal{S}) \quad = \quad \mathrm{disc}(A)$$

Theorem (Lovász, Spencer & Vesztergombi '86)

Finding good colorings $\Rightarrow \forall x \in [0,1]^m$ *can find* $y \in \{0,1\}^m$ *with* $\|Ax - Ay\|_\infty$ *small*

Entropy rounding (simple version)

- **Input:** $A \in \mathbb{R}^{n \times m}, x \in [0,1]^m$
- **Assume:** $\forall$ submatrix $A' \subseteq A$: half-coloring χ: $\|A'\chi\|_\infty \leq \Delta$
- **Output:** $y \in \{0,1\}^m$: $\|Ax - Ay\|_\infty \leq O(\log m) \cdot \Delta$

(1) $y := x$

(2) FOR *phase* $k =$ last bit TO 1 DO

  (3) Call y_i active if kth bit is 1

  (4) Find half-coloring χ : active var $\rightarrow \{-1, +1, 0\}$

  (5) Update $y' := y + (\frac{1}{2})^k \chi$

  (6) REPEAT WHILE $\exists$ active var.

$$A = \begin{pmatrix} \boxed{} & \boxed{} & \boxed{} & \boxed{} \end{pmatrix}$$

$$y = (\, 0.11 \; 0.10 \; 0.11 \; 0.01 \,)$$

Entropy rounding (simple version)

- **Input:** $A \in \mathbb{R}^{n \times m}, x \in [0,1]^m$
- **Assume:** $\forall$ submatrix $A' \subseteq A$: half-coloring χ: $\|A'\chi\|_\infty \leq \Delta$
- **Output:** $y \in \{0,1\}^m$: $\|Ax - Ay\|_\infty \leq O(\log m) \cdot \Delta$

(1) $y := x$

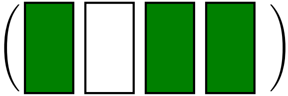
(2) FOR *phase* k = last bit TO 1 DO

  (3) Call y_i active if kth bit is 1

  (4) Find half-coloring χ : active var $\to \{-1, +1, 0\}$

  (5) Update $y' := y + (\frac{1}{2})^k \chi$

  (6) REPEAT WHILE $\exists$ active var.

$A =$ 

$y = (0.1\mathbf{1}\ 0.10\ 0.1\mathbf{1}\ 0.0\mathbf{1})$

Entropy rounding (simple version)

- **Input:** $A \in \mathbb{R}^{n \times m}, x \in [0,1]^m$
- **Assume:** $\forall$ submatrix $A' \subseteq A$: half-coloring χ: $\|A'\chi\|_\infty \leq \Delta$
- **Output:** $y \in \{0,1\}^m$: $\|Ax - Ay\|_\infty \leq O(\log m) \cdot \Delta$

(1) $y := x$

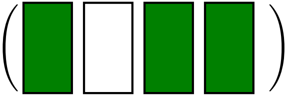
(2) FOR *phase* $k =$ last bit TO 1 DO

  (3) Call y_i active if kth bit is 1

  (4) Find half-coloring χ : active var $\to \{-1, +1, 0\}$

  (5) Update $y' := y + (\frac{1}{2})^k \chi$

  (6) REPEAT WHILE $\exists$ active var.

$A =$ 

$y = (\, 0.11 \;\; 0.10 \;\; 0.11 \;\; 0.01 \,)$

$\chi = (\, +1 \qquad\qquad -1 \quad 0 \,)$

Entropy rounding (simple version)

- **Input:** $A \in \mathbb{R}^{n \times m}, x \in [0,1]^m$
- **Assume:** $\forall$ submatrix $A' \subseteq A$: half-coloring χ: $\|A'\chi\|_\infty \leq \Delta$
- **Output:** $y \in \{0,1\}^m$: $\|Ax - Ay\|_\infty \leq O(\log m) \cdot \Delta$

(1) $y := x$

(2) FOR *phase* k = last bit TO 1 DO

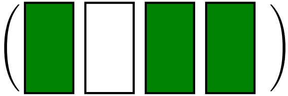
  (3) Call y_i active if kth bit is 1

  (4) Find half-coloring χ : active var $\to \{-1, +1, 0\}$

  (5) Update $y' := y + (\frac{1}{2})^k \chi$

  (6) REPEAT WHILE $\exists$ active var.

$A =$



$y = (\,0.11\ \ 0.10\ \ 0.11\ \ 0.01\,)$

$\chi = (\ \ +1 \qquad\qquad -1 \quad\ 0\ \ )$

$y' = (\,1.00\ \ 0.10\ \ 0.10\ \ 0.01\,)$

Entropy rounding (simple version)

- **Input:** $A \in \mathbb{R}^{n\times m}, x \in [0,1]^m$
- **Assume:** $\forall$ submatrix $A' \subseteq A$: half-coloring χ: $\|A'\chi\|_\infty \leq \Delta$
- **Output:** $y \in \{0,1\}^m$: $\|Ax - Ay\|_\infty \leq O(\log m)\cdot\Delta$

(1) $y := x$

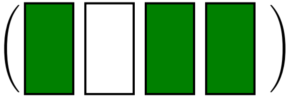
(2) FOR *phase* $k =$ last bit TO 1 DO

  (3) Call y_i active if kth bit is 1

  (4) Find half-coloring χ : active var $\to \{-1, +1, 0\}$

  (5) Update $y' := y + (\frac{1}{2})^k \chi$

  (6) REPEAT WHILE $\exists$ active var.

$A =$ 

$$y = (\,0.1\mathbf{1}\ \ 0.10\ \ 0.1\mathbf{1}\ \ 0.0\mathbf{1}\,)$$

$$\chi = (\ \ +1\ \ \ \ \ \ \ \ \ \ \ \ -1\ \ \ \ 0\ \ )$$

$$y' = (\,1.00\ \ 0.10\ \ 0.10\ \ 0.0\mathbf{1}\,)$$

Analysis:

- A phase has $\leq \log m$ iterations

Entropy rounding (simple version)

- **Input:** $A \in \mathbb{R}^{n \times m}, x \in [0,1]^m$
- **Assume:** $\forall$ submatrix $A' \subseteq A$: half-coloring χ: $\|A'\chi\|_\infty \leq \Delta$
- **Output:** $y \in \{0,1\}^m$: $\|Ax - Ay\|_\infty \leq O(\log m) \cdot \Delta$

(1) $y := x$

(2) FOR *phase* $k =$ last bit TO 1 DO

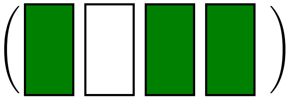
  (3) Call y_i active if kth bit is 1

  (4) Find half-coloring χ : active var $\to \{-1, +1, 0\}$

  (5) Update $y' := y + (\frac{1}{2})^k \chi$

  (6) REPEAT WHILE $\exists$ active var.

$A =$



$y = (\,0.11\ 0.10\ 0.11\ 0.01\,)$

$\chi = (\ +1 \qquad\quad -1 \quad 0\ )$

$y' = (\,1.00\ 0.10\ 0.10\ 0.01\,)$

Analysis:

- A phase has $\leq \log m$ iterations
- During phase k: $\|Ay' - Ay\|_\infty = (\frac{1}{2})^k \|A\chi\|_\infty \leq (\frac{1}{2})^k \Delta$

Entropy rounding (simple version)

- **Input:** $A \in \mathbb{R}^{n\times m}, x \in [0,1]^m$
- **Assume:** $\forall$ submatrix $A' \subseteq A$: half-coloring χ: $\|A'\chi\|_\infty \leq \Delta$
- **Output:** $y \in \{0,1\}^m$: $\|Ax - Ay\|_\infty \leq O(\log m)\cdot\Delta$

(1) $y := x$

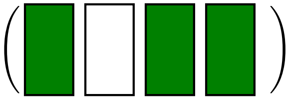
(2) FOR *phase* k = last bit TO 1 DO

  (3) Call y_i active if kth bit is 1

  (4) Find half-coloring χ : active var $\to \{-1, +1, 0\}$

  (5) Update $y' := y + (\frac{1}{2})^k \chi$

  (6) REPEAT WHILE $\exists$ active var.

$A =$ 

$$y = (\,0.11\ \ 0.10\ \ 0.11\ \ 0.01\,)$$

$$\chi = (\ +1\qquad\quad -1\quad\ 0\ \ )$$

$$y' = (\,1.00\ \ 0.10\ \ 0.10\ \ 0.01\,)$$

Analysis:

- A phase has $\leq \log m$ iterations
- During phase k: $\|Ay' - Ay\|_\infty = (\frac{1}{2})^k \|A\chi\|_\infty \leq (\frac{1}{2})^k \Delta$
- Triangle inequality:

$$\|Ax - Ay\|_\infty \leq \sum_{k\geq 1} \sum_{t=1}^{\log m} \left(\frac{1}{2}\right)^k \cdot \Delta = O(\log m)\cdot\Delta \qquad \square$$

Entropy

Definition

For random variable Z, the **entropy** is

$$H(Z) = \sum_z \Pr[Z = z] \cdot \log_2 \left(\frac{1}{\Pr[Z = z]} \right)$$

Entropy

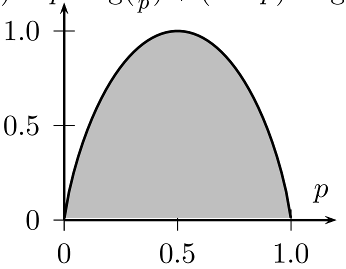
Definition

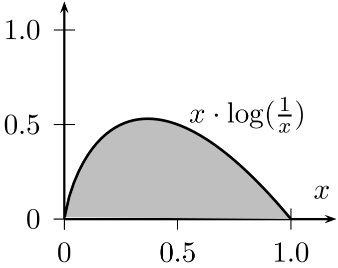
For random variable Z, the **entropy** is

$$H(Z) = \sum_z \Pr[Z = z] \cdot \log_2 \left(\frac{1}{\Pr[Z = z]} \right)$$

Example: $\Pr[Z = a] = p$ and $\Pr[Z = b] = 1 - p$

$H(Z) = p \cdot \log(\frac{1}{p}) + (1 - p) \cdot \log(\frac{1}{1-p})$





Entropy

Definition

For random variable Z, the **entropy** is

$$H(Z) = \sum_z \Pr[Z = z] \cdot \log_2 \left(\frac{1}{\Pr[Z = z]} \right)$$

Properties:

- *Uniform distribution maximizes entropy:* If Z attains k distinct values: $H(Z) \leq \log_2(k)$ (attained if $\Pr[Z = z] = \frac{1}{k} \; \forall z$).

Entropy

Definition

For random variable Z, the **entropy** is

$$H(Z) = \sum_z \Pr[Z = z] \cdot \log_2 \left(\frac{1}{\Pr[Z = z]} \right)$$

Properties:

- *Uniform distribution maximizes entropy:* If Z attains k distinct values: $H(Z) \leq \log_2(k)$ (attained if $\Pr[Z = z] = \frac{1}{k}\ \forall z$).
- *One likely event:* $\exists z : \Pr[Z = z] \geq (\frac{1}{2})^{H(Z)}$

Entropy

Definition

For random variable Z, the **entropy** is

$$H(Z) = \sum_z \Pr[Z = z] \cdot \log_2 \left(\frac{1}{\Pr[Z = z]} \right)$$

Properties:

- *Uniform distribution maximizes entropy:* If Z attains k distinct values: $H(Z) \leq \log_2(k)$ (attained if $\Pr[Z = z] = \frac{1}{k} \; \forall z$).
- *One likely event:* $\exists z : \Pr[Z = z] \geq (\frac{1}{2})^{H(Z)}$
- *Subadditivity:* $H(f(Z, Z')) \leq H(Z) + H(Z')$.

Chernov-type bound

Lemma

Let $X_1, \ldots, X_n$ *be indep. RV with* $\Pr[X_i = \pm 1] = \frac{1}{2}$.

$$\Pr\left[\Big|\sum_{i=1}^{n} X_i\Big| \geq \lambda\sqrt{n}\right] \leq 2e^{-\lambda^2/2}.$$

Chernov-type bound

Lemma

Let $X_1, \ldots, X_n$ *be indep. RV with* $\Pr[X_i = \pm\alpha_i] = \frac{1}{2}$.

$$\Pr\left[\Big|\sum_{i=1}^{n} X_i\Big| \geq \lambda\|\alpha\|_2\right] \leq 2e^{-\lambda^2/2}.$$

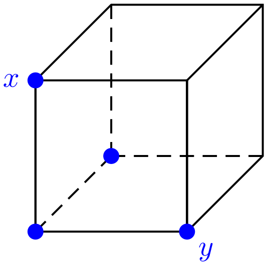
- **Standard deviation**:

$$\sqrt{\mathrm{Var}[\sum_i X_i]} = \sqrt{\sum_i E[(X_i - E[X_i])^2]} = \sqrt{\sum_{i=1}^{n} \alpha_i^2} = \|\alpha\|_2$$

An isoperimetric inequality

Lemma (Special case of Isoperimetric Ineq – Kleitman'66)

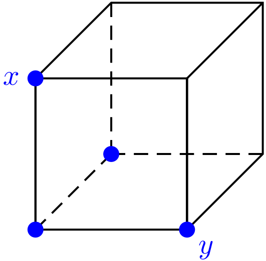
For any $X \subseteq \{0,1\}^n$ *of size* $|X| \geq 2^{0.8n}$ *and* $n \geq 2$, *there are* $x, y \in X$ *with* $\|x - y\|_1 \geq n/2$.



An isoperimetric inequality

Lemma (Special case of Isoperimetric Ineq – Kleitman'66)

For any $X \subseteq \{0,1\}^n$ of size $|X| \geq 2^{0.8n}$ and $n \geq 2$, there are $x, y \in X$ with $\|x - y\|_1 \geq n/2$.

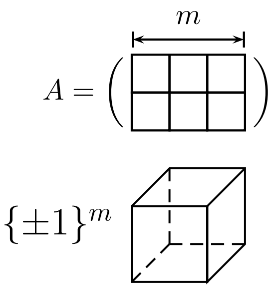


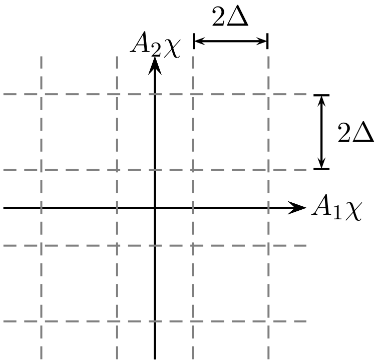
- Proof with weaker constant:

$$\left| \begin{array}{c} \text{ball of radius } n/10 \\ \text{around } \mathbf{0} \end{array} \right| \leq \sum_{0 \leq q < n/10} \binom{n}{q} \leq \left(\frac{en}{n/10}\right)^{n/10} < 2^{0.8n}$$

Theorem [Beck's entropy method]

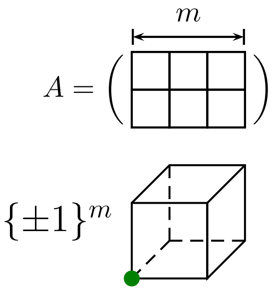
$$\underset{\chi_i \in \{\pm 1\}}{H}\left(\left\lceil \frac{A\chi}{2\Delta} \right\rfloor\right) \leq \frac{m}{5} \Rightarrow \exists \text{half-coloring } \chi^0 : \|A\chi^0\|_\infty \leq \Delta.$$

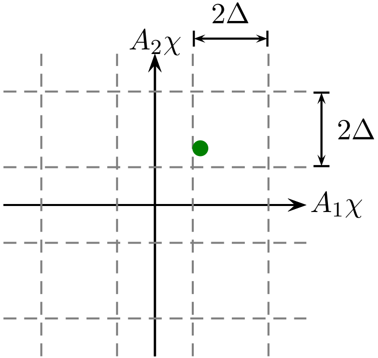




Theorem [Beck's entropy method]

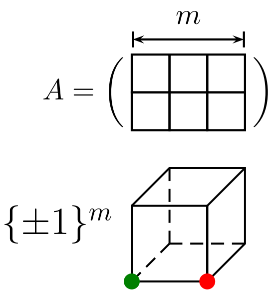
$$\underset{\chi_i \in \{\pm 1\}}{H}\left(\left\lceil \frac{A\chi}{2\Delta} \right\rfloor\right) \leq \frac{m}{5} \Rightarrow \exists \text{half-coloring } \chi^0 : \|A\chi^0\|_\infty \leq \Delta.$$

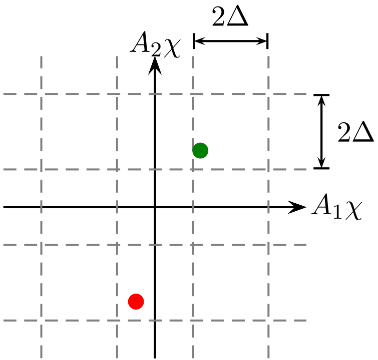




Theorem [Beck's entropy method]

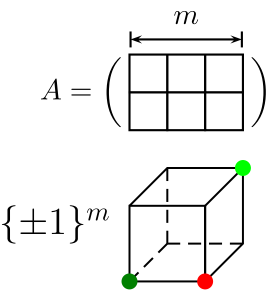
$$\underset{\chi_i \in \{\pm 1\}}{H}\left(\left\lceil \frac{A\chi}{2\Delta}\right\rfloor\right) \leq \frac{m}{5} \Rightarrow \exists\text{half-coloring } \chi^0 : \|A\chi^0\|_\infty \leq \Delta.$$

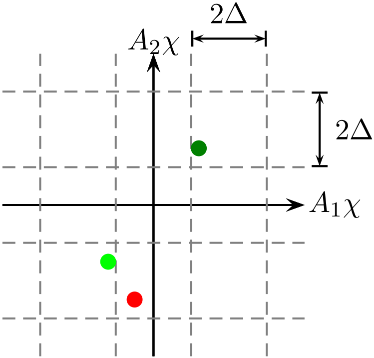




Theorem [Beck's entropy method]

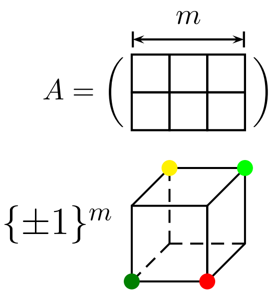
$$\underset{\chi_i\in\{\pm 1\}}{H}\left(\left\lceil \frac{A\chi}{2\Delta}\right\rfloor\right) \le \frac{m}{5} \Rightarrow \exists\text{half-coloring } \chi^0 : \|A\chi^0\|_\infty \le \Delta.$$

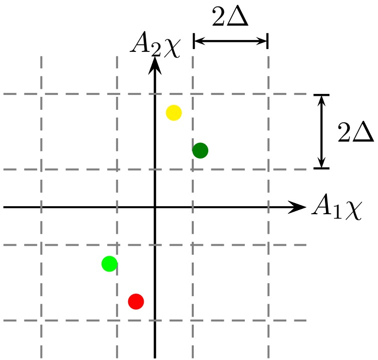




Theorem [Beck's entropy method]

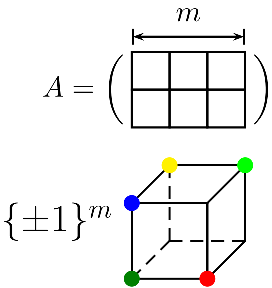
$$\underset{\chi_i \in \{\pm 1\}}{H}\left(\left\lceil \frac{A\chi}{2\Delta} \right\rfloor\right) \leq \frac{m}{5} \Rightarrow \exists \text{half-coloring } \chi^0 : \|A\chi^0\|_\infty \leq \Delta.$$

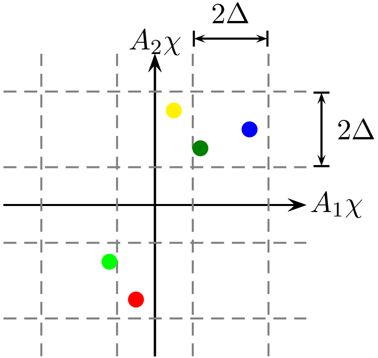




Theorem [Beck's entropy method]

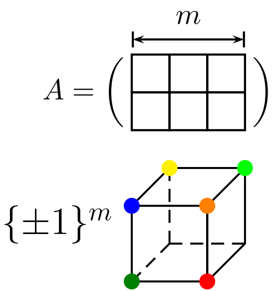
$$\underset{\chi_i \in \{\pm 1\}}{H}\left(\left\lceil \frac{A\chi}{2\Delta} \right\rfloor\right) \leq \frac{m}{5} \Rightarrow \exists\text{half-coloring } \chi^0 : \|A\chi^0\|_\infty \leq \Delta.$$

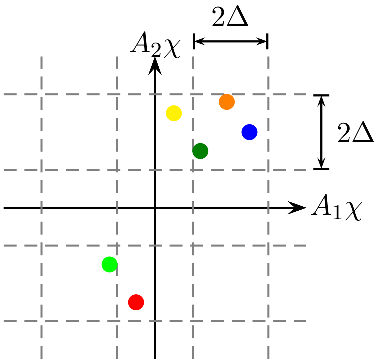




Theorem [Beck's entropy method]

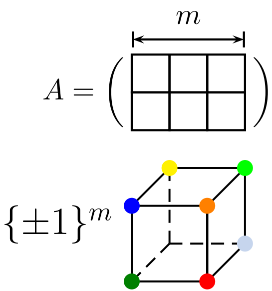
$$\underset{\chi_i \in \{\pm 1\}}{H}\left(\left\lceil \frac{A\chi}{2\Delta} \right\rceil\right) \leq \frac{m}{5} \Rightarrow \exists \text{half-coloring } \chi^0 : \|A\chi^0\|_\infty \leq \Delta.$$

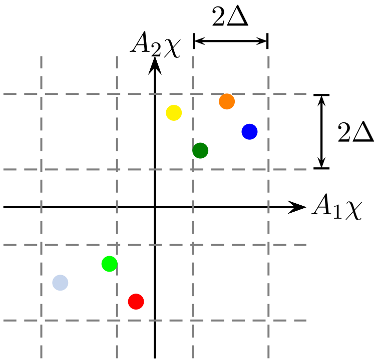




Theorem [Beck's entropy method]

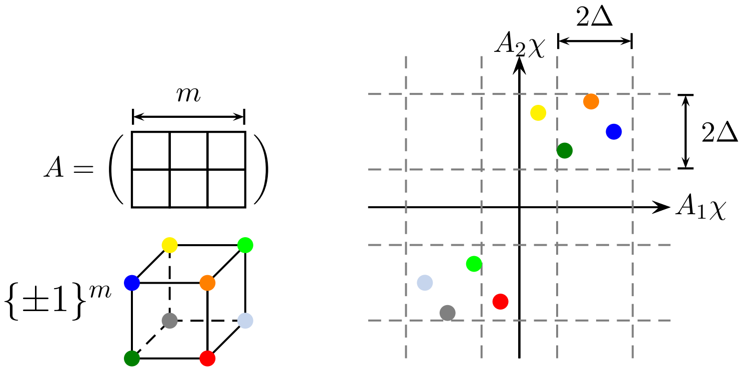
$$\underset{\chi_i \in \{\pm 1\}}{H}\left(\left\lceil \frac{A\chi}{2\Delta} \right\rceil\right) \leq \frac{m}{5} \Rightarrow \exists \text{half-coloring } \chi^0 : \|A\chi^0\|_\infty \leq \Delta.$$





Theorem [Beck's entropy method]

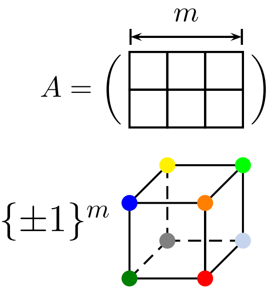
$$\underset{\chi_i \in \{\pm 1\}}{H}\left(\left\lceil \frac{A\chi}{2\Delta} \right\rfloor\right) \leq \frac{m}{5} \Rightarrow \exists \text{half-coloring } \chi^0 : \|A\chi^0\|_\infty \leq \Delta.$$

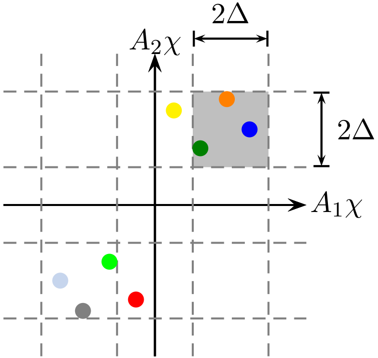


Theorem [Beck's entropy method]

$$\underset{\chi_i \in \{\pm 1\}}{H}\left(\left\lceil \frac{A\chi}{2\Delta} \right\rfloor\right) \leq \frac{m}{5} \Rightarrow \exists \text{half-coloring } \chi^0 : \|A\chi^0\|_\infty \leq \Delta.$$

- $\exists$cell : $\Pr[A\chi \in \text{cell}] \geq (\frac{1}{2})^{m/5}$.

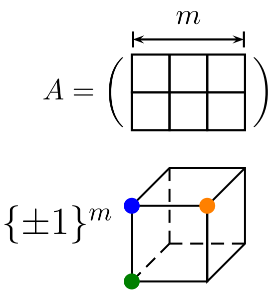


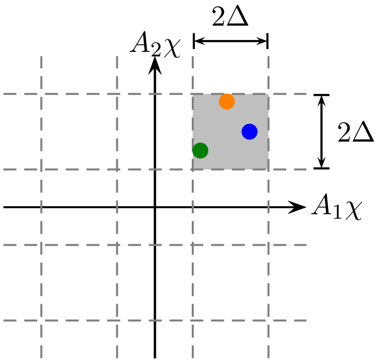


Theorem [Beck's entropy method]

$$\underset{\chi_i \in \{\pm 1\}}{H}\left(\left\lceil \frac{A\chi}{2\Delta} \right\rfloor\right) \leq \frac{m}{5} \Rightarrow \exists \text{half-coloring } \chi^0 : \|A\chi^0\|_\infty \leq \Delta.$$

- $\exists$cell : $\Pr[A\chi \in \text{cell}] \geq (\frac{1}{2})^{m/5}$.
- At least $2^m \cdot (\frac{1}{2})^{m/5} = 2^{0.8m}$ colorings χ have $A\chi \in$ cell

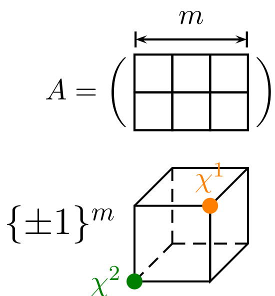


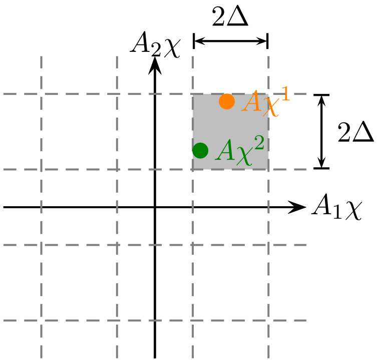


Theorem [Beck's entropy method]

$$\underset{\chi_i \in \{\pm 1\}}{H}\left(\left\lceil \frac{A\chi}{2\Delta} \right\rfloor\right) \leq \frac{m}{5} \Rightarrow \exists \text{half-coloring } \chi^0 : \|A\chi^0\|_\infty \leq \Delta.$$

- $\exists$cell : $\Pr[A\chi \in \text{cell}] \geq (\frac{1}{2})^{m/5}$.
- At least $2^m \cdot (\frac{1}{2})^{m/5} = 2^{0.8m}$ colorings χ have $A\chi \in$ cell
- Pick χ^1, χ^2 differing in half of entries

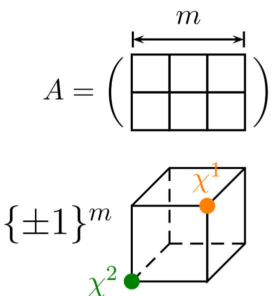


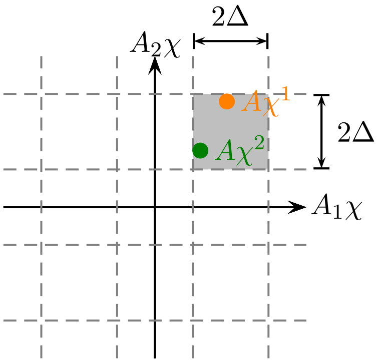


Theorem [Beck's entropy method]

$$\underset{\chi_i \in \{\pm 1\}}{H}\left(\left\lceil \frac{A\chi}{2\Delta} \right\rfloor\right) \leq \frac{m}{5} \Rightarrow \exists\text{half-coloring } \chi^0 : \|A\chi^0\|_\infty \leq \Delta.$$

- $\exists$cell : $\Pr[A\chi \in \text{cell}] \geq (\frac{1}{2})^{m/5}$.
- At least $2^m \cdot (\frac{1}{2})^{m/5} = 2^{0.8m}$ colorings χ have $A\chi \in$ cell
- Pick χ^1, χ^2 differing in half of entries
- Define $\chi^0(i) := \frac{1}{2}(\chi^1(i) - \chi^2(i)) \in \{0, \pm 1\}$.

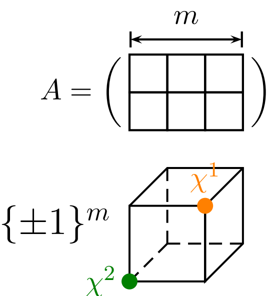


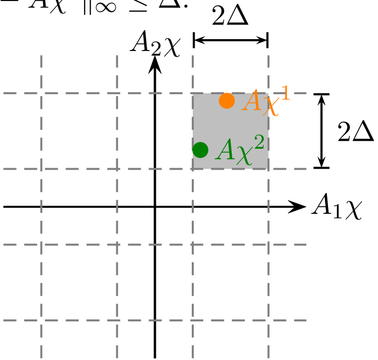


Theorem [Beck's entropy method]

$$\underset{\chi_i \in \{\pm 1\}}{H}\left(\left\lceil \frac{A\chi}{2\Delta} \right\rfloor\right) \leq \frac{m}{5} \Rightarrow \exists \text{half-coloring } \chi^0 : \|A\chi^0\|_\infty \leq \Delta.$$

- $\exists$cell : $\Pr[A\chi \in \text{cell}] \geq (\frac{1}{2})^{m/5}$.
- At least $2^m \cdot (\frac{1}{2})^{m/5} = 2^{0.8m}$ colorings χ have $A\chi \in$ cell
- Pick χ^1, χ^2 differing in half of entries
- Define $\chi^0(i) := \frac{1}{2}(\chi^1(i) - \chi^2(i)) \in \{0, \pm 1\}$.
- Then $\|A\chi^0\|_\infty \leq \frac{1}{2}\|A\chi^1 - A\chi^2\|_\infty \leq \Delta$. $\square$



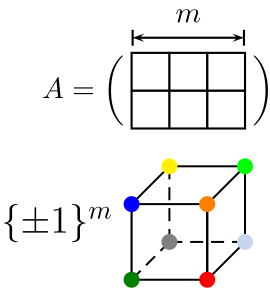


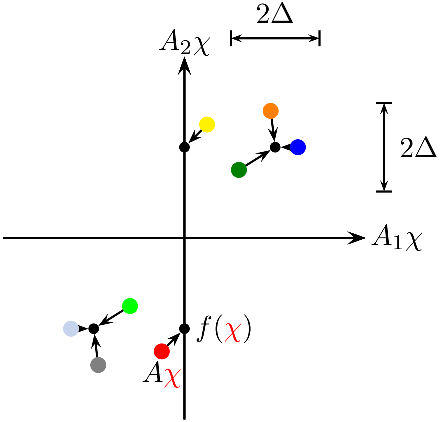
A slight generalization

Theorem

For any auxiliary function $f(\chi)$ with $\|A\chi - f(\chi)\|_\infty \leq \Delta$:

$$\underset{\chi_i \in \{\pm 1\}}{H}(f(\chi)) \leq \frac{m}{5} \Rightarrow \exists \text{half-coloring } \chi^0 : \|A\chi^0\|_\infty \leq \Delta.$$



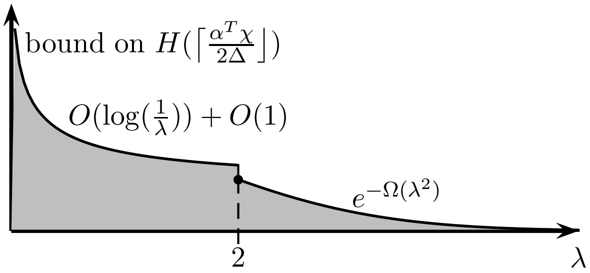


A bound on the entropy

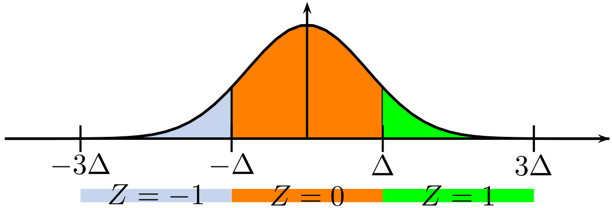
Lemma

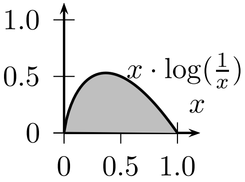
For $\alpha \in \mathbb{R}^m,\ \Delta > 0$:

$$\underset{\chi_i \in \{\pm 1\}}{H}\left(\left\lceil \frac{\alpha^T \chi}{2\Delta} \right\rfloor\right) \leq G\Big(\underbrace{\frac{\Delta}{\|\alpha\|_2}}_{=:\lambda}\Big) := \begin{cases} 9e^{-\lambda^2/5} & \textit{if } \lambda \geq 2 \\ \log_2(32 + 64/\lambda) & \textit{if } \lambda < 2 \end{cases}$$



Proof – Case $\lambda \geq 2$

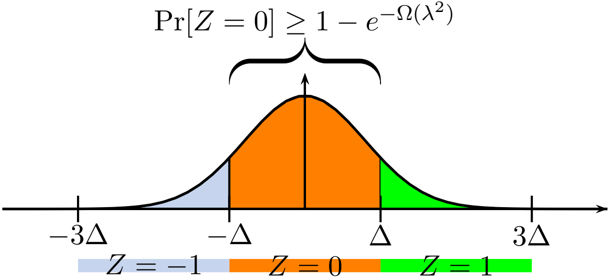


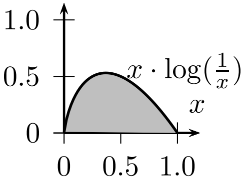


- Recall: $\Delta = \lambda \cdot \|\alpha\|_2$ with $\lambda \geq 2$ and $Z = \left\lceil \frac{\alpha^T \chi}{2\Delta} \right\rfloor$

$$H(Z) = \sum_{i \in \mathbb{Z}} \Pr[Z = i] \cdot \log\left(\frac{1}{\Pr[Z = i]}\right)$$

Proof – Case $\lambda \geq 2$



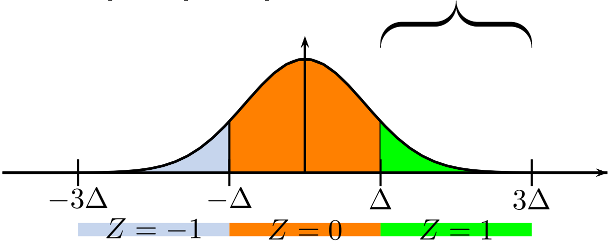


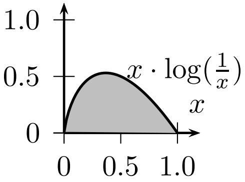
- Recall: $\Delta = \lambda \cdot \|\alpha\|_2$ with $\lambda \geq 2$ and $Z = \left\lceil \frac{\alpha^T \chi}{2\Delta} \right\rceil$
- $\Pr[Z=0] \cdot \log(\frac{1}{\Pr[Z=0]}) \leq e^{-\Omega(\lambda^2)}$

$$H(Z) = \sum_{i \in \mathbb{Z}} \Pr[Z=i] \cdot \log\left(\frac{1}{\Pr[Z=i]}\right)$$

Proof – Case $\lambda \geq 2$

$$\Pr[Z = i] \leq \Pr[\alpha^T \chi \text{ is } \geq i\lambda \text{ times standard dev}] \leq e^{-\Omega(i^2\lambda^2)}$$



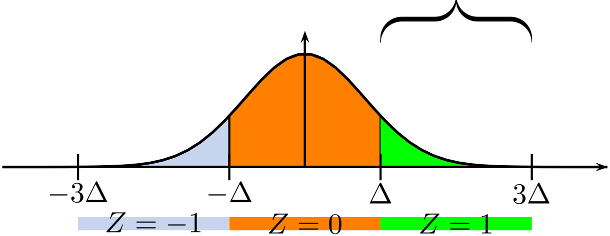


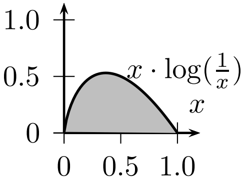
- Recall: $\Delta = \lambda \cdot \|\alpha\|_2$ with $\lambda \geq 2$ and $Z = \left\lceil \frac{\alpha^T \chi}{2\Delta} \right\rfloor$
- $\Pr[Z = 0] \cdot \log(\frac{1}{\Pr[Z=0]}) \leq e^{-\Omega(\lambda^2)}$
- $\Pr[Z = i] \cdot \log(\frac{1}{\Pr[Z=i]}) \leq e^{-\Omega(i^2\lambda^2)} \cdot \log(e^{i^2\lambda^2})$

$$H(Z) = \sum_{i \in \mathbb{Z}} \Pr[Z = i] \cdot \log\left(\frac{1}{\Pr[Z = i]}\right)$$

Proof – Case $\lambda \geq 2$

$$\Pr[Z = i] \leq \Pr[\alpha^T \chi \text{ is } \geq i\lambda \text{ times standard dev}] \leq e^{-\Omega(i^2\lambda^2)}$$

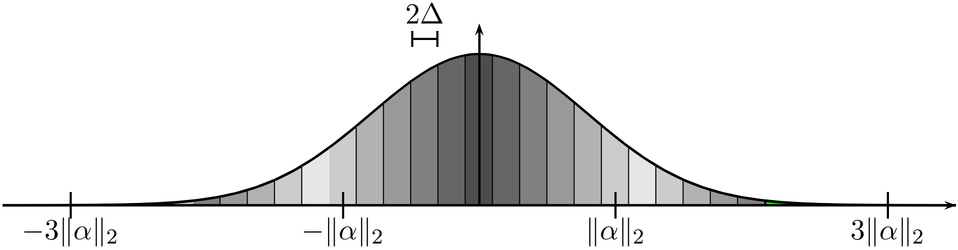




- Recall: $\Delta = \lambda \cdot \|\alpha\|_2$ with $\lambda \geq 2$ and $Z = \left\lceil \frac{\alpha^T \chi}{2\Delta} \right\rfloor$
- $\Pr[Z = 0] \cdot \log(\frac{1}{\Pr[Z=0]}) \leq e^{-\Omega(\lambda^2)}$
- $\Pr[Z = i] \cdot \log(\frac{1}{\Pr[Z=i]}) \leq e^{-\Omega(i^2\lambda^2)} \cdot \log(e^{i^2\lambda^2})$

$$H(Z) = \sum_{i \in \mathbb{Z}} \Pr[Z = i] \cdot \log\left(\frac{1}{\Pr[Z = i]}\right) \leq e^{-\Omega(\lambda^2)}$$

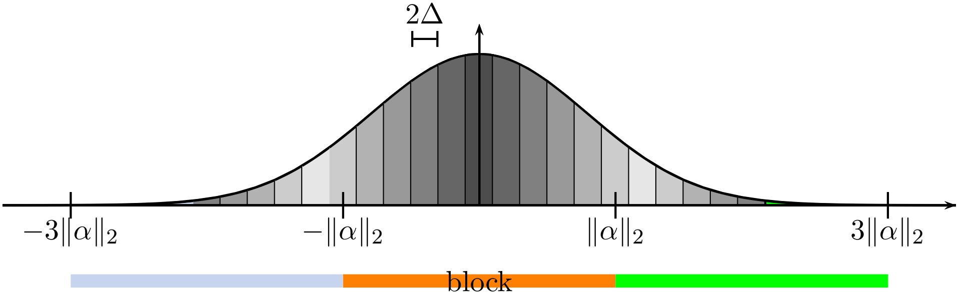
Proof – Case $\lambda < 2$



Subadditivity:

$$H(Z) \quad \leq$$

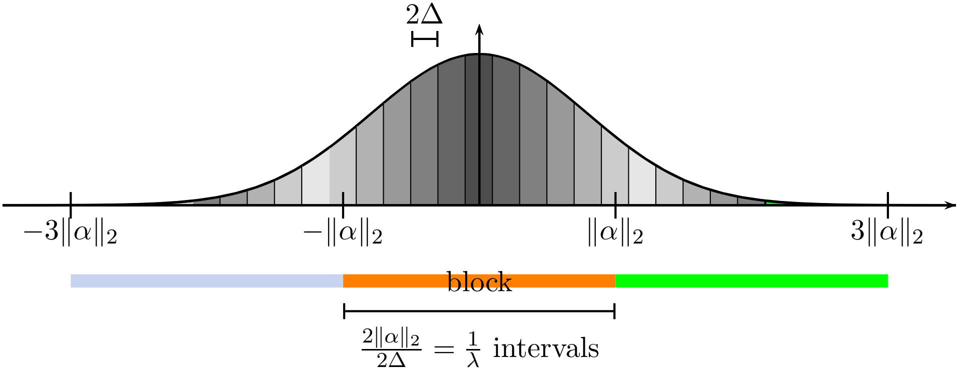
Proof – Case $\lambda < 2$



Subadditivity:

$$\begin{aligned} H(Z) &\leq H(\text{which block of length } 2\|\alpha\|_2) + \\ &\leq O(1) + \end{aligned}$$

Proof – Case $\lambda < 2$



Subadditivity:

$$\begin{aligned} H(Z) &\leq H(\text{which block of length } 2\|\alpha\|_2) + H(\text{index}) \\ &\leq O(1) + O(\log \tfrac{1}{\lambda}) \quad \square \end{aligned}$$

Entropy rounding (extended version)

Algorithm:

- Input: $A \in \mathbb{R}^{n\times m}, x \in [0,1]^m$

(1) $y := x$

(2) FOR *phase* $k =$ last bit TO 1 DO

 (3) Call y_i active if kth bit is 1

 (4) Find half-coloring χ : active var $\to \{-1, +1, 0\}$

 (5) Update $y' := y + (\frac{1}{2})^k \chi$

 (6) REPEAT WHILE $\exists$ active var.

Entropy rounding (extended version)

Algorithm:

- Input: $A \in [-1,1]^{n\times m}, x \in [0,1]^m$
- Row weights $w(i)$ ($\sum_i w(i) = 1;\ w(i) \geq 0$)

(1) $y := x$

(2) FOR *phase* $k =$ last bit TO 1 DO

  (3) Call y_i active if kth bit is 1

  (4) Find half-coloring χ : active var $\rightarrow \{-1, +1, 0\}$ with $|A_i\chi| \leq \Delta_i$ s.t. $G(\frac{\Delta_i}{\sqrt{\#\text{active var}}}) \leq w(i) \cdot \frac{\#\text{active var}}{5}$

  (5) Update $y' := y + (\frac{1}{2})^k \chi$

  (6) REPEAT WHILE $\exists$ active var.

Entropy rounding (extended version)

Algorithm:

- Input: $A \in [-1,1]^{n\times m}, x \in [0,1]^m$
- Row weights $w(i)$ ($\sum_i w(i) = 1;\ w(i) \geq 0$)

(1) $y := x$

(2) FOR *phase* $k =$ last bit TO 1 DO

 (3) Call y_i active if kth bit is 1

 (4) Find half-coloring χ : active var $\to \{-1, +1, 0\}$
 with $|A_i\chi| \leq \Delta_i$ s.t. $G(\frac{\Delta_i}{\sqrt{\#\text{active var}}}) \leq w(i) \cdot \frac{\#\text{active var}}{5}$

 (5) Update $y' := y + (\frac{1}{2})^k\chi$

 (6) REPEAT WHILE $\exists$ active var.

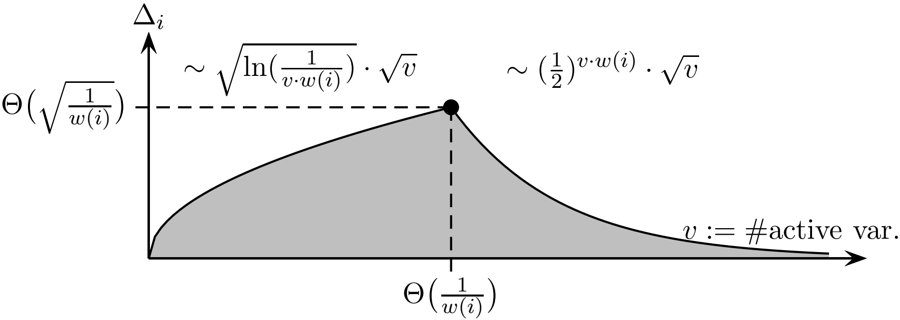
- In each step:

$$H\left(\left(\left\lceil \frac{A_i\chi}{2\Delta_i}\right\rfloor\right)_i\right) \overset{\text{Subadd.}}{\leq} \sum_{i=1}^{n} H\left(\left\lceil \frac{A_i\chi}{2\Delta_i}\right\rfloor\right) \leq \sum_{i=1}^{n} G\left(\frac{\Delta_i}{\sqrt{\#\text{act. var}}}\right) \leq \frac{\#\text{act.var.}}{5}$$

- Use $\alpha \in [-1,1]^{m'} \Rightarrow \|\alpha\|_2 \leq \sqrt{m'}$

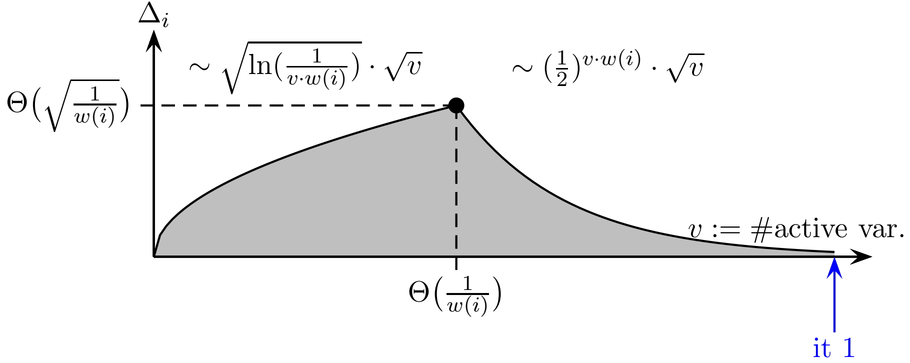
Entropy rounding (extended version) (2)

- Consider row i while rounding bit k:



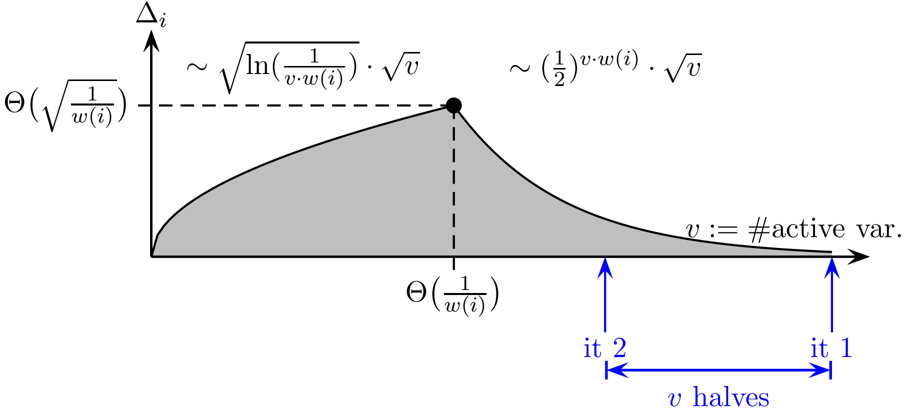
Entropy rounding (extended version) (2)

- Consider row i while rounding bit k:



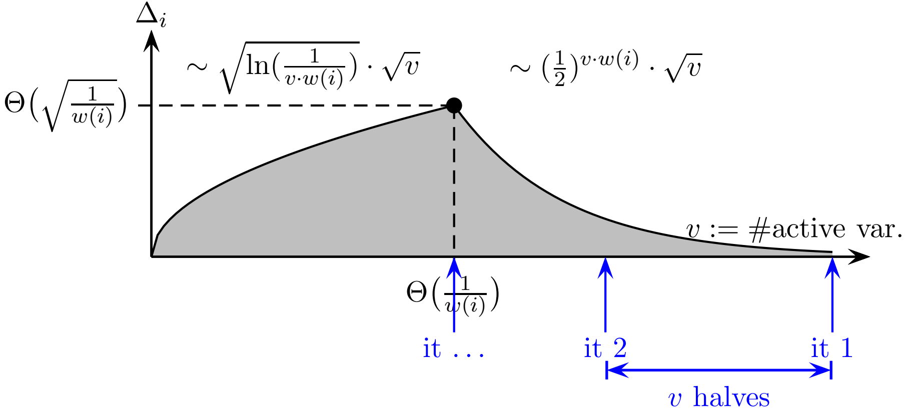
Entropy rounding (extended version) (2)

- Consider row i while rounding bit k:



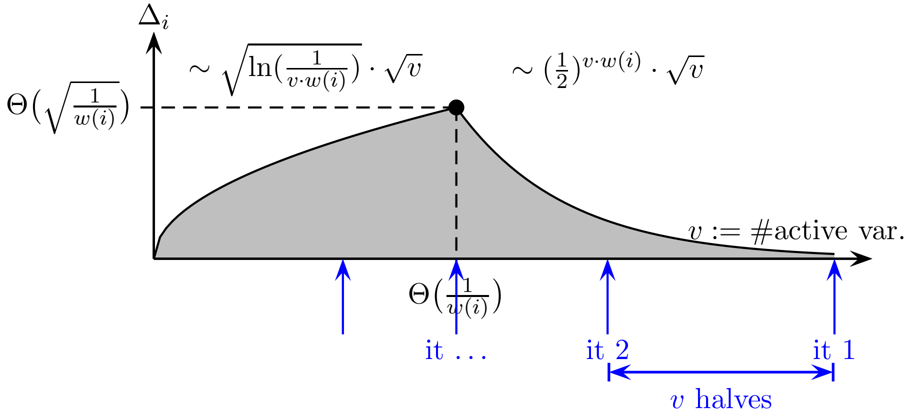
Entropy rounding (extended version) (2)

- Consider row i while rounding bit k:



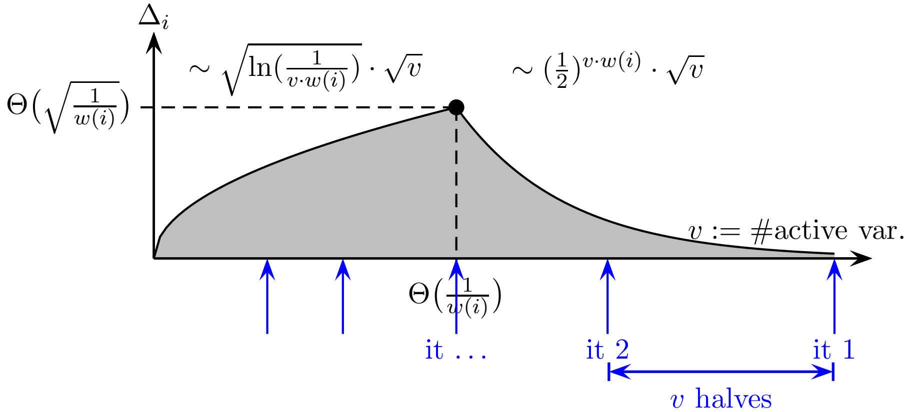
Entropy rounding (extended version) (2)

- Consider row i while rounding bit k:



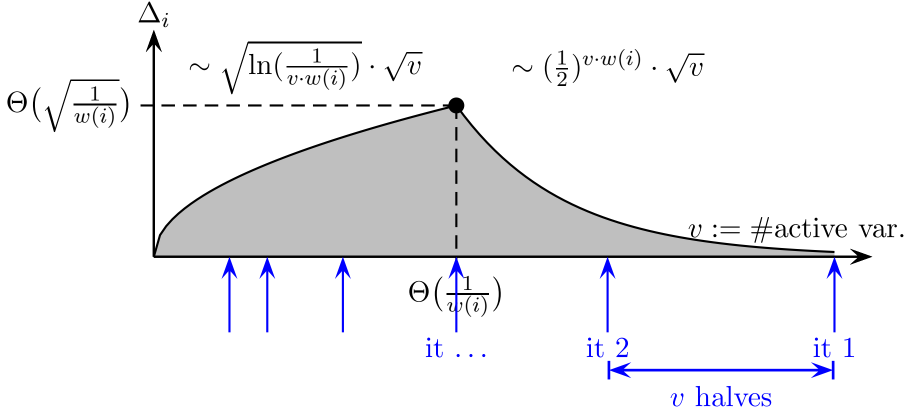
Entropy rounding (extended version) (2)

- Consider row i while rounding bit k:



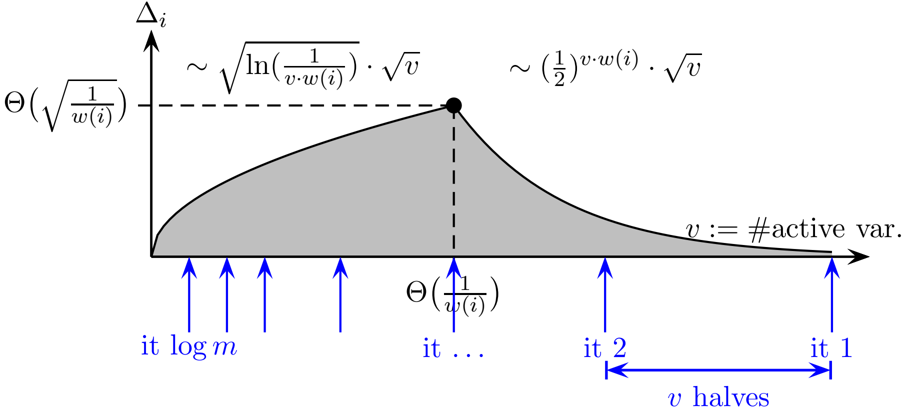
Entropy rounding (extended version) (2)

- Consider row i while rounding bit k:



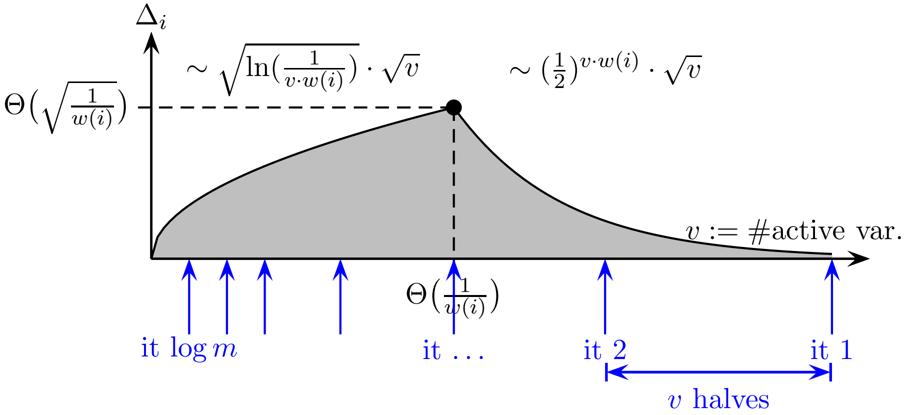
Entropy rounding (extended version) (2)

- Consider row i while rounding bit k:



Entropy rounding (extended version) (2)

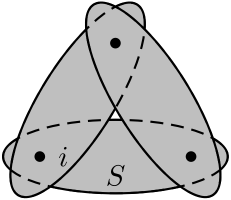
- Consider row i while rounding bit k:



- By convergence: $|A_i x - A_i y| \leq O\left(\sqrt{\frac{1}{w(i)}}\right)$

Example: Discrepancy of set systems

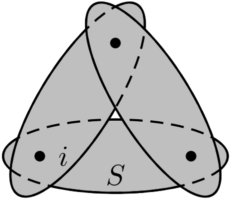
- **Given:** Set system $\mathcal{S}$ with n sets and n elements



$$A = \begin{array}{cc} \begin{pmatrix}1&1&0\\0&1&1\\1&0&1\end{pmatrix} & \begin{array}{l} \\ \text{set } S \\ \\ \end{array} \\ \begin{array}{ccc} & i & \end{array} & \end{array}$$

Example: Discrepancy of set systems

- **Given:** Set system $\mathcal{S}$ with n sets and n elements

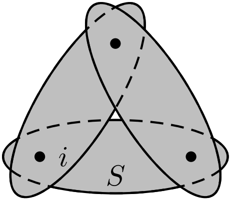


$$A = \begin{array}{cc} \begin{pmatrix}1&1&0\\0&1&1\\1&0&1\end{pmatrix} & \begin{array}{l} \\ \text{set } S \\ \\ \end{array} \\ \begin{array}{ccc} & i & \end{array} & \end{array}$$

- Pick $x := (\frac{1}{2}, \dots, \frac{1}{2})$; weight $w(i) := \frac{1}{n}$ for each row

Example: Discrepancy of set systems

- **Given:** Set system $\mathcal{S}$ with n sets and n elements

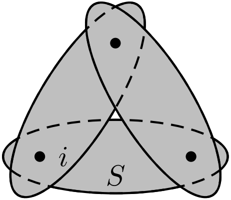


$$A = \begin{array}{cl} \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix} & \begin{array}{l} \\ \text{set } S \\ \\ \end{array} \\ \begin{array}{ccc} & i & \end{array} & \end{array}$$

- Pick $x := (\frac{1}{2}, \ldots, \frac{1}{2})$; weight $w(i) := \frac{1}{n}$ for each row
- There is $y \in \{0,1\}^n : \|Ax - Ay\|_\infty = O(\sqrt{\frac{1}{1/n}}) = O(\sqrt{n})$.

Example: Discrepancy of set systems

- **Given:** Set system $\mathcal{S}$ with n sets and n elements



$$A = \begin{array}{cc} \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix} & \begin{array}{l} \\ \text{set } S \\ \\ \end{array} \\ \begin{array}{ccc} & i & \end{array} & \end{array}$$

- Pick $x := (\frac{1}{2}, \ldots, \frac{1}{2})$; weight $w(i) := \frac{1}{n}$ for each row
- There is $y \in \{0,1\}^n : \|Ax - Ay\|_\infty = O(\sqrt{\frac{1}{1/n}}) = O(\sqrt{n})$.
- Coloring $\chi(i) = \begin{cases} +1 & y_i = 1 \\ -1 & y_i = 0 \end{cases}$ has discrepancy $O(\sqrt{n})$.
- "*6 Standard deviations suffice*"-Thm [Spencer '85]

Summarizing

Theorem

Input:

- *matrix* $A \in [-1,1]^{n \times m}$ *(*$\forall A' \subseteq A$*: there are* $f : -\Delta \leq A'\chi - f(\chi) \leq \Delta$ *and* $H(f(\chi)) \leq \frac{\#cols(A')}{10}$*)*
- *vector* $x \in [0,1]^m$
- *row weights* $w(i)$ *(*$\sum_i w(i) = 1$*)*

There is a $y \in \{0,1\}^m$ *with*

- *Bounded difference:*
  - $|A_i x - A_i y| \leq O(\log(m)) \cdot \Delta_i$
  - $|A_i x - A_i y| \leq O(\sqrt{1/w(i)})$

Summarizing

Theorem

Input:

- *matrix* $A \in [-1,1]^{n \times m}$ *(*$\forall A' \subseteq A$*: there are* $f : -\Delta \leq A'\chi - f(\chi) \leq \Delta$ *and* $H(f(\chi)) \leq \frac{\#cols(A')}{10}$*)*
- *vector* $x \in [0,1]^m$
- *row weights* $w(i)$ *(*$\sum_i w(i) = 1$*)*

There is a random variable $y \in \{0,1\}^m$ *with*

- *Bounded difference:*
  - $|A_i x - A_i y| \leq O(\log(m)) \cdot \Delta_i$
  - $|A_i x - A_i y| \leq O(\sqrt{1/w(i)})$
- *Preserved expectation:* $E[y_i] = x_i$.

- Randomness: $y = x + \sum_{k \geq 1} \sum_{t=1}^{\log m} (\frac{1}{2})^k \cdot \chi^{(k,t)}$

Summarizing

Theorem

Input:

- *matrix* $A \in [-1,1]^{n \times m}$ *(*$\forall A' \subseteq A$*: there are* $f : -\Delta \leq A'\chi - f(\chi) \leq \Delta$ *and* $H(f(\chi)) \leq \frac{\#cols(A')}{10}$*)*
- *vector* $x \in [0,1]^m$
- *row weights* $w(i)$ *(*$\sum_i w(i) = 1$*)*

There is a random variable $y \in \{0,1\}^m$ *with*

- *Bounded difference:*
  - $|A_i x - A_i y| \leq O(\log(m)) \cdot \Delta_i$
  - $|A_i x - A_i y| \leq O(\sqrt{1/w(i)})$
- *Preserved expectation:* $E[y_i] = x_i$*.*

- Randomness: $y = x + \sum_{k \geq 1} \sum_{t=1}^{\log m} (\frac{1}{2})^k \cdot \textbf{rand.} \pm \mathbf{1} \cdot \chi^{(k,t)}$

Summarizing

Theorem

Input:

- *matrix* $A \in [-1,1]^{n\times m}$ *(*$\forall A' \subseteq A$*: there are* $f : -\Delta \leq A'\chi - f(\chi) \leq \Delta$ *and* $H(f(\chi)) \leq \frac{\#cols(A')}{10}$*)*
- *vector* $x \in [0,1]^m$
- *row weights* $w(i)$ *(*$\sum_i w(i) = 1$*)*

There is a random variable $y \in \{0,1\}^m$ *with*

- *Bounded difference:*
  - $|A_i x - A_i y| \leq O(\log(m)) \cdot \Delta_i$
  - $|A_i x - A_i y| \leq O(\sqrt{1/w(i)})$
- *Preserved expectation:* $E[y_i] = x_i$*.*

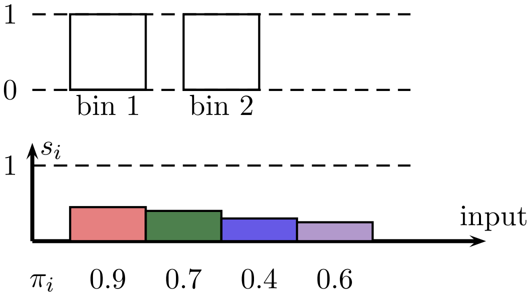
- Randomness: $y = x + \sum_{k\geq 1} \sum_{t=1}^{\log m} (\frac{1}{2})^k \cdot \textbf{rand.} \pm \mathbf{1} \cdot \chi^{(k,t)}$
- Can be computed by SDP in poly-time using [Bansal '10]

Bin Packing With Rejection

Input:

- Items $i \in \{1, \ldots, n\}$ with **size** $s_i \in [0, 1]$, and **rejection penalty** $\pi_i \in [0, 1]$

Goal: Pack or reject. Minimize # **bins** + rejection cost.

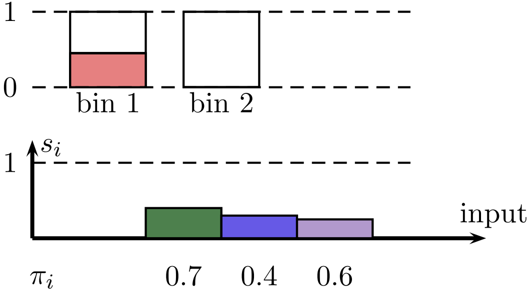


Bin Packing With Rejection

Input:

- Items $i \in \{1, \ldots, n\}$ with **size** $s_i \in [0, 1]$, and **rejection penalty** $\pi_i \in [0, 1]$

Goal: Pack or reject. Minimize # **bins** + rejection cost.

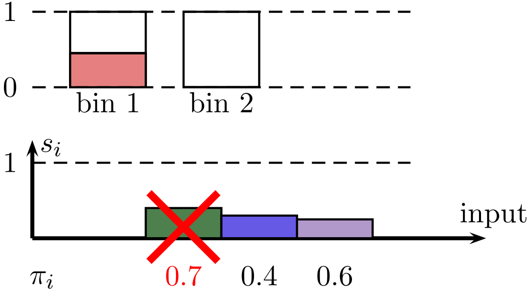


Bin Packing With Rejection

Input:

- Items $i \in \{1, \ldots, n\}$ with **size** $s_i \in [0, 1]$, and **rejection penalty** $\pi_i \in [0, 1]$

Goal: Pack or reject. Minimize # **bins** + rejection cost.

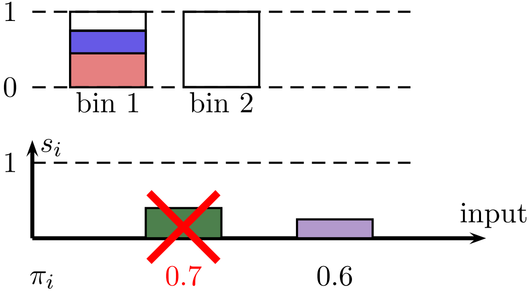


Bin Packing With Rejection

Input:

- Items $i \in \{1, \ldots, n\}$ with **size** $s_i \in [0, 1]$, and **rejection penalty** $\pi_i \in [0, 1]$

Goal: Pack or reject. Minimize # **bins** + rejection cost.

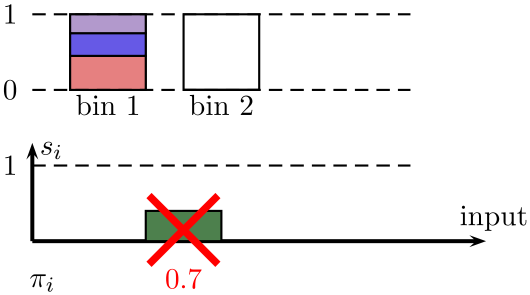


Bin Packing With Rejection

Input:

- Items $i \in \{1, \dots, n\}$ with **size** $s_i \in [0, 1]$, and **rejection penalty** $\pi_i \in [0, 1]$

Goal: Pack or reject. Minimize # **bins** + rejection cost.



Known results

Bin Packing With Rejection:

- APTAS [Epstein '06]
- Faster APTAS [Bein, Correa & Han '08]
- AFPTAS ($APX \leq OPT + \frac{OPT}{(\log OPT)^{1-o(1)}}$)
  [Epstein & Levin '10]

Known results

Bin Packing With Rejection:

- APTAS [Epstein '06]
- Faster APTAS [Bein, Correa & Han '08]
- AFPTAS ($APX \leq OPT + \frac{OPT}{(\log OPT)^{1-o(1)}}$) [Epstein & Levin '10]

Bin Packing:

- $APX \leq OPT + O(\log^2 OPT)$ [Karmarkar & Karp '82]

Known results

Bin Packing With Rejection:

- APTAS [Epstein '06]
- Faster APTAS [Bein, Correa & Han '08]
- AFPTAS ($APX \leq OPT + \frac{OPT}{(\log OPT)^{1-o(1)}}$) [Epstein & Levin '10]

Bin Packing:

- $APX \leq OPT + O(\log^2 OPT)$ [Karmarkar & Karp '82]

Theorem

There is a randomized approximation algorithm for **Bin Packing With Rejection** with

$$APX \leq OPT + O(\log^2 OPT)$$

(with high probability).

The column-based LP

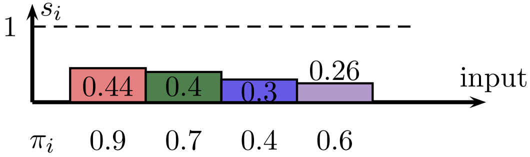
Set Cover formulation:

- **Bins:** Sets $S \subseteq [n]$ with $\sum_{i \in S} s_i \leq 1$ of cost $c(S) = 1$
- **Rejections:** Sets $S = \{i\}$ of cost $c(S) = \pi_i$

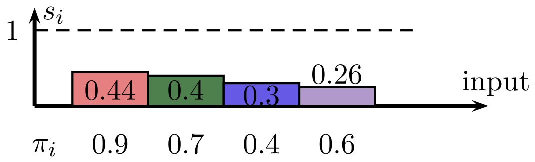
LP:

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

The column-based LP - Example



The column-based LP - Example

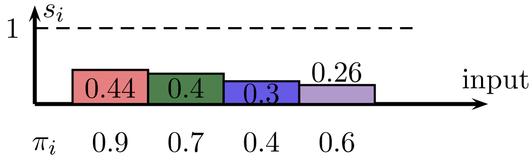


$$\min \left(1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \;\middle|\; .9 \quad .7 \quad .4 \quad .6\right) x$$

$$\left(\begin{array}{cccccccccccc|cccc} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{array}\right) x \geq \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

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

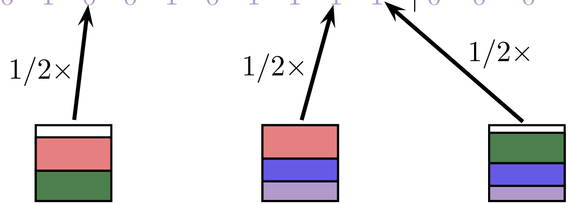
The column-based LP - Example



$$\min \left(1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1 \;\middle|\; .9\ .7\ .4\ .6\right) x$$

$$\left(\begin{array}{cccccccccccc|cccc}
1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1
\end{array}\right) x \geq \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

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



Massaging the LP

- Sort items $s_1 \geq \ldots \geq s_n$

Massaging the LP

- Sort items $s_1 \geq \ldots \geq s_n$
- Call items **large** if size at least $\varepsilon := \frac{\log OPT_f}{OPT_f}$ (otherwise **small**

Massaging the LP

- Sort items $s_1 \geq \ldots \geq s_n$
- Call items **large** if size at least $\varepsilon := \frac{\log OPT_f}{OPT_f}$ (otherwise **small**

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

Massaging the LP

- Sort items $s_1 \geq \ldots \geq s_n$
- Call items **large** if size at least $\varepsilon := \frac{\log OPT_f}{OPT_f}$ (otherwise **small**
- Add up rows $1, \ldots, i$ of constraint matrix to obtain row i for matrix A (for large i's).
  "1 slot per item" $\Rightarrow$ "i slots for largest i items"

$$\begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \end{pmatrix} \to A = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 0 \\ 2 & 1 & 1 & 1 & 1 & 0 \\ 2 & 2 & 2 & 1 & 1 & 1 \\ & & & & & \\ & & & & & \end{pmatrix}$$

Massaging the LP

- Sort items $s_1 \geq \ldots \geq s_n$
- Call items **large** if size at least $\varepsilon := \frac{\log OPT_f}{OPT_f}$ (otherwise **small**
- Add up rows $1, \ldots, i$ of constraint matrix to obtain row i for matrix A (for large i's).
  "1 slot per item" $\Rightarrow$ "i slots for largest i items"
- Append row vector $\left(\sum_{i \in S : s_i < \varepsilon} s_i\right)_S$

$$\begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \end{pmatrix} \to A = \begin{pmatrix} \boxed{\begin{matrix} 1 & 0 & 1 & 1 & 0 & 0 \\ 2 & 1 & 1 & 1 & 1 & 0 \\ 2 & 2 & 2 & 1 & 1 & 1 \end{matrix}} \\ \boxed{\text{space for small items}} \end{pmatrix}$$

Massaging the LP

- Sort items $s_1 \geq \ldots \geq s_n$
- Call items **large** if size at least $\varepsilon := \frac{\log OPT_f}{OPT_f}$ (otherwise **small**
- Add up rows $1, \ldots, i$ of constraint matrix to obtain row i for matrix A (for large i's).
  "1 slot per item" $\Rightarrow$ "i slots for largest i items"
- Append row vector $(\sum_{i \in S : s_i < \varepsilon} s_i)_S$
- Append objective function c

$$\begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \end{pmatrix} \to A = \begin{pmatrix} \boxed{\begin{matrix} 1 & 0 & 1 & 1 & 0 & 0 \\ 2 & 1 & 1 & 1 & 1 & 0 \\ 2 & 2 & 2 & 1 & 1 & 1 \end{matrix}} \\ \boxed{\text{space for small items}} \\ \boxed{\text{objective function}} \end{pmatrix}$$

Entropy bound for monotone matrices

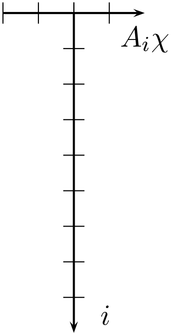
Theorem

Let A be column-monotone matrix, max entry $\leq \Delta$, sum of last row $= \sigma$. There are auxiliary RV f: $\|A\chi - f(\chi)\|_\infty = O(\Delta)$ and $H_\chi(f(\chi)) \leq O(\frac{\sigma}{\Delta})$.

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 2 & 0 \\ 1 & 1 & 2 & 0 \\ 2 & 1 & 2 & 0 \\ 2 & 1 & 2 & 1 \\ 2 & 1 & 2 & 2 \\ 2 & 1 & 3 & 2 \end{pmatrix}$$

$\sigma = \sum$ (last row)

$\leq \Delta$



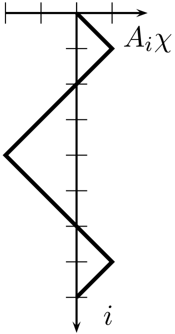
Entropy bound for monotone matrices

Theorem

Let A be column-monotone matrix, max entry $\leq \Delta$, sum of last row $= \sigma$. There are auxiliary RV f: $\|A\chi - f(\chi)\|_\infty = O(\Delta)$ and $H_\chi(f(\chi)) \leq O(\frac{\sigma}{\Delta})$.

$$\chi = (+1,-1,-1,+1)$$

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 2 & 0 \\ 1 & 1 & 2 & 0 \\ 2 & 1 & 2 & 0 \\ 2 & 1 & 2 & 1 \\ 2 & 1 & 2 & 2 \\ 2 & 1 & 3 & 2 \end{pmatrix}$$

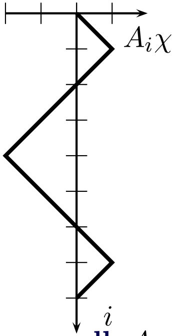


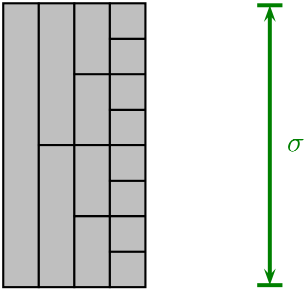
- **Idea:** Describe **random walk** $A_1\chi, \ldots, A_\sigma\chi$ $O(\Delta)$-approximately

Entropy bound for monotone matrices

$\chi = (+1,-1,-1,+1)$

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 2 & 0 \\ 1 & 1 & 2 & 0 \\ 2 & 1 & 2 & 0 \\ 2 & 1 & 2 & 1 \\ 2 & 1 & 2 & 2 \\ 2 & 1 & 3 & 2 \end{pmatrix}$$



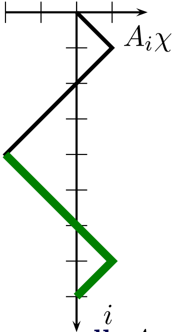


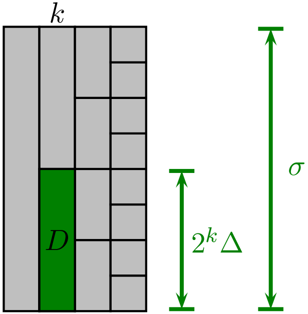
- **Idea:** Describe **random walk** $A_1\chi, \ldots, A_\sigma\chi$ $O(\Delta)$-approximately

Entropy bound for monotone matrices

$\chi = (+1,-1,-1,+1)$

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 2 & 0 \\ 1 & 1 & 2 & 0 \\ 2 & 1 & 2 & 0 \\ 2 & 1 & 2 & 1 \\ 2 & 1 & 2 & 2 \\ 2 & 1 & 3 & 2 \end{pmatrix}$$



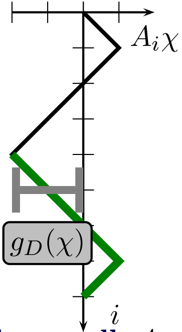


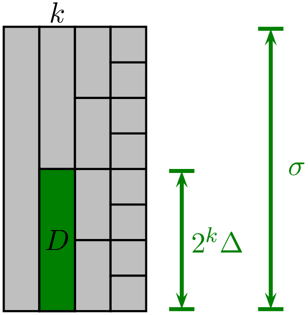
- **Idea:** Describe **random walk** $A_1\chi, \dots, A_\sigma\chi$ $O(\Delta)$-approximately
- For each interval D of length $2^k \cdot \Delta$:
  $g_D(\chi) :=$ covered distance in D rounded to $\frac{\Delta}{1.1^k}$

Entropy bound for monotone matrices

$\chi = (+1, -1, -1, +1)$

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 2 & 0 \\ 1 & 1 & 2 & 0 \\ 2 & 1 & 2 & 0 \\ 2 & 1 & 2 & 1 \\ 2 & 1 & 2 & 2 \\ 2 & 1 & 3 & 2 \end{pmatrix}$$



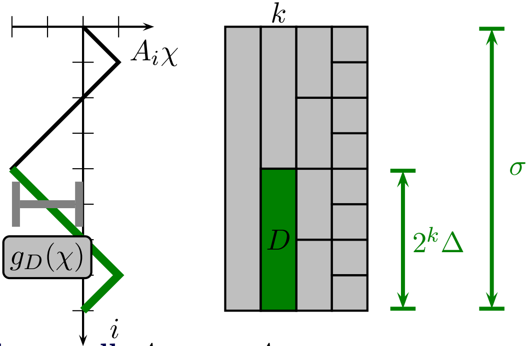


- **Idea:** Describe **random walk** $A_1\chi, \ldots, A_\sigma\chi$ $O(\Delta)$-approximately
- For each interval D of length $2^k \cdot \Delta$: $g_D(\chi) :=$ covered distance in D rounded to $\frac{\Delta}{1.1^k}$

Entropy bound for monotone matrices

$\chi = (+1,-1,-1,+1)$

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 2 & 0 \\ 1 & 1 & 2 & 0 \\ 2 & 1 & 2 & 0 \\ 2 & 1 & 2 & 1 \\ 2 & 1 & 2 & 2 \\ 2 & 1 & 3 & 2 \end{pmatrix}$$

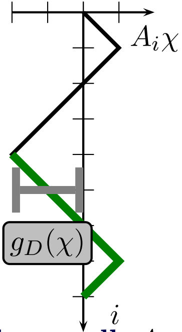


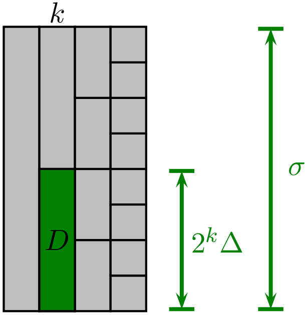
- **Idea:** Describe **random walk** $A_1\chi, \dots, A_\sigma\chi$ $O(\Delta)$-approximately
- For each interval D of length $2^k \cdot \Delta$: $g_D(\chi) :=$ covered distance in D rounded to $\frac{\Delta}{1.1^k}$
- Std-Dev for g_D: $\sqrt{\text{max dep.} \cdot |D|} \leq \sqrt{\Delta \cdot 2^k \Delta} = 2^{k/2} \cdot \Delta$

Entropy bound for monotone matrices

$\chi = (+1, -1, -1, +1)$

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 2 & 0 \\ 1 & 1 & 2 & 0 \\ 2 & 1 & 2 & 0 \\ 2 & 1 & 2 & 1 \\ 2 & 1 & 2 & 2 \\ 2 & 1 & 3 & 2 \end{pmatrix}$$



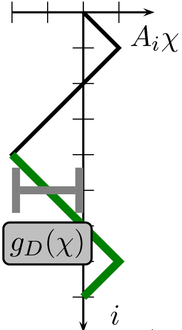


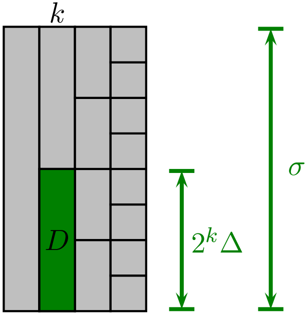
- **Idea:** Describe **random walk** $A_1\chi, \dots, A_\sigma\chi$ $O(\Delta)$-approximately
- For each interval D of length $2^k \cdot \Delta$: $g_D(\chi) :=$ covered distance in D rounded to $\frac{\Delta}{1.1^k}$
- Std-Dev for g_D: $\sqrt{\text{max dep.} \cdot |D|} \le \sqrt{\Delta \cdot 2^k \Delta} = 2^{k/2} \cdot \Delta$
- $H(g_D) \le G(\frac{\Delta/1.1^k}{2^{k/2}\Delta}) \le G(2^{-k}) = O(\log 2^k) = O(k)$

Entropy bound for monotone matrices

$\chi = (+1, -1, -1, +1)$

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 2 & 0 \\ 1 & 1 & 2 & 0 \\ 2 & 1 & 2 & 0 \\ 2 & 1 & 2 & 1 \\ 2 & 1 & 2 & 2 \\ 2 & 1 & 3 & 2 \end{pmatrix}$$



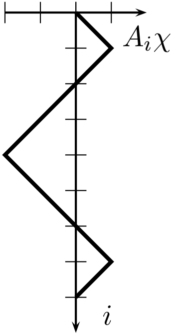


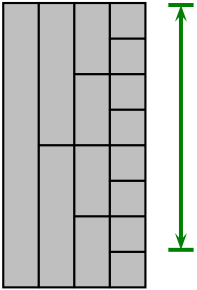
- **Idea:** Describe **random walk** $A_1\chi, \ldots, A_\sigma\chi$ $O(\Delta)$-approximately
- For each interval D of length $2^k \cdot \Delta$: $g_D(\chi) :=$ covered distance in D rounded to $\frac{\Delta}{1.1^k}$
- Std-Dev for g_D: $\sqrt{\text{max dep.} \cdot |D|} \leq \sqrt{\Delta \cdot 2^k \Delta} = 2^{k/2} \cdot \Delta$
- $H(g_D) \leq G(\frac{\Delta/1.1^k}{2^{k/2}\Delta}) \leq G(2^{-k}) = O(\log 2^k) = O(k)$
- Total entropy of g: $\sum_{k \geq 1} \frac{\sigma}{2^k\Delta} \cdot O(k) = O(\frac{\sigma}{\Delta})$.

Entropy bound for monotone matrices

$$\chi = (+1,-1,-1,+1)$$

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 2 & 0 \\ 1 & 1 & 2 & 0 \\ 2 & 1 & 2 & 0 \\ 2 & 1 & 2 & 1 \\ 2 & 1 & 2 & 2 \\ 2 & 1 & 3 & 2 \end{pmatrix}$$



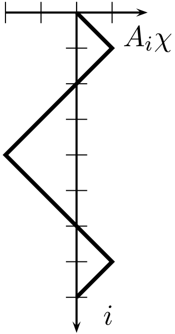


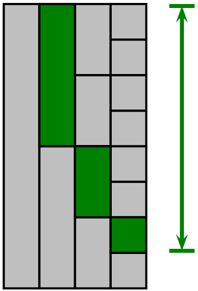
- We know $A_i\chi$ up to an error of:

Entropy bound for monotone matrices

$$\chi = (+1, -1, -1, +1)$$

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 2 & 0 \\ 1 & 1 & 2 & 0 \\ 2 & 1 & 2 & 0 \\ 2 & 1 & 2 & 1 \\ 2 & 1 & 2 & 2 \\ 2 & 1 & 3 & 2 \end{pmatrix}$$



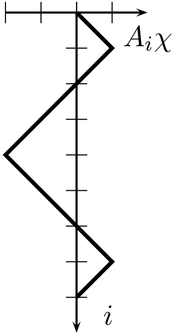


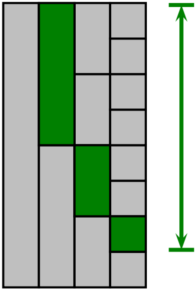
- We know $A_i\chi$ up to an error of:

Entropy bound for monotone matrices

$\chi = ({\color{blue}+1},{\color{red}-1},{\color{red}-1},{\color{blue}+1})$

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 2 & 0 \\ 1 & 1 & 2 & 0 \\ 2 & 1 & 2 & 0 \\ 2 & 1 & 2 & 1 \\ 2 & 1 & 2 & 2 \\ 2 & 1 & 3 & 2 \end{pmatrix}$$





- We know $A_i\chi$ up to an error of: $\sum_{k\geq 1} \frac{\Delta}{1.1^k} = O(\Delta)$.
  (formally $f_i(\chi) := \sum_{D:\dot{\bigcup} D=[i]} g_D(\chi)$) $\square$

Approximation algorithm for BPWR

- Apply rounding theorem to fractional sol. x

$$A = \left(\begin{array}{c} \boxed{\begin{matrix} 1 & 0 & 1 & 1 & 0 & 0 \\ 2 & 1 & 1 & 1 & 1 & 0 \\ 2 & 2 & 2 & 1 & 1 & 1 \end{matrix}} \\ \boxed{\text{space for small items}} \\ \boxed{\text{objective function}} \end{array}\right) \begin{array}{l} \\ \\ \\ \text{weight } 1/2 \\ \text{weight } 1/2 \end{array}$$

Approximation algorithm for BPWR

- Apply rounding theorem to fractional sol. x
- Pick $\Delta_i := \Theta(\frac{1}{s_i})$

$$A = \begin{array}{r} \\ i \\ \\ \\ \\ \end{array}\begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 0 \\ 2 & 1 & 1 & 1 & 1 & 0 \\ 2 & 2 & 2 & 1 & 1 & 1 \\ \multicolumn{6}{c}{\text{space for small items}} \\ \multicolumn{6}{c}{\text{objective function}} \end{pmatrix} \begin{array}{l} \\ \Delta_i := \Theta(\frac{1}{s_i}) \\ \\ \text{weight } 1/2 \\ \text{weight } 1/2 \end{array}$$

Approximation algorithm for BPWR

- Apply rounding theorem to fractional sol. x
- Pick $\Delta_i := \Theta(\frac{1}{s_i})$
- Assume for 1 sec that $\frac{1}{2k} \leq s_i \leq \frac{1}{k}$ (same **size class**)

$$O\left(\frac{\sigma}{\Delta}\right) \leq \frac{1}{20} \sum_{S \text{ active}} |S| \cdot \frac{1}{k} \leq \frac{1}{10} \sum_{S \text{ active}} \underbrace{\sum_{i \in S} s_i}_{\leq 1} \leq \frac{\#\text{ active var.}}{10}$$

$$A = \begin{array}{c} \\ i \\ \\ \\ \\ \end{array} \begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 0 \\ 2 & 1 & 1 & 1 & 1 & 0 \\ 2 & 2 & 2 & 1 & 1 & 1 \\ \multicolumn{6}{c}{\text{space for small items}} \\ \multicolumn{6}{c}{\text{objective function}} \end{pmatrix} \begin{array}{l} \\ \Delta_i := \Theta(\frac{1}{s_i}) \\ \\ \text{weight } 1/2 \\ \text{weight } 1/2 \end{array}$$

Approximation algorithm for BPWR (2)

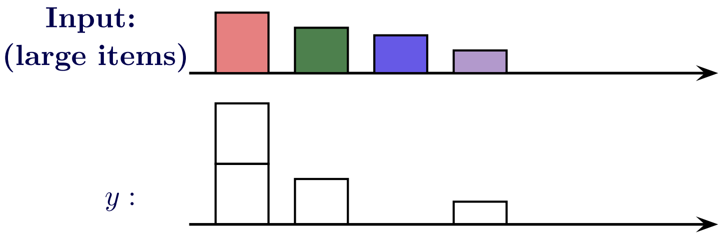
Obtain $y \in \{0,1\}^m$:

- $|A_i y - A_i x| \leq O(\log m) \cdot \frac{1}{s_i}$
- $|c^T x - c^T y| \leq O(1)$
- space for small items in x and y differs by $O(1)$

Approximation algorithm for BPWR (2)

Obtain $y \in \{0,1\}^m$: $\rightarrow$ repair to feasible solution

- $A_i y \geq A_i x = i$ (y reserves $\geq i$ slots for largest i items)
  Buy $O(\log m) \cdot \frac{1}{s_i}$ slots for each size class $\rightarrow O(\log m \cdot \log \frac{1}{\varepsilon})$
- $|c^T x - c^T y| \leq O(1)$
- space for small items in x and y differs by $O(1)$

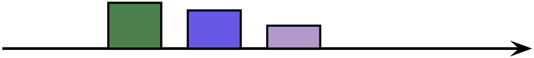


Approximation algorithm for BPWR (2)

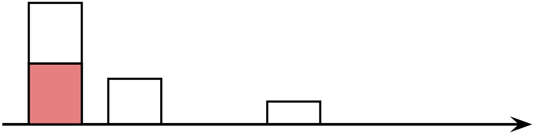
Obtain $y \in \{0,1\}^m$: $\rightarrow$ repair to feasible solution

- $A_i y \geq A_i x = i$ (y reserves $\geq i$ slots for largest i items)
  Buy $O(\log m) \cdot \frac{1}{s_i}$ slots for each size class $\rightarrow O(\log m \cdot \log \frac{1}{\varepsilon})$
- $|c^T x - c^T y| \leq O(1)$
- space for small items in x and y differs by $O(1)$

Input:
(large items)



$y:$

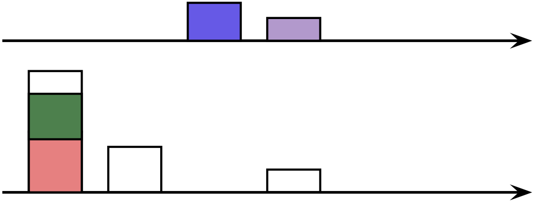


Approximation algorithm for BPWR (2)

Obtain $y \in \{0,1\}^m$: $\rightarrow$ repair to feasible solution

- $A_i y \geq A_i x = i$ (y reserves $\geq i$ slots for largest i items)
  Buy $O(\log m) \cdot \frac{1}{s_i}$ slots for each size class $\rightarrow O(\log m \cdot \log \frac{1}{\varepsilon})$
- $|c^T x - c^T y| \leq O(1)$
- space for small items in x and y differs by $O(1)$

Input:
(large items)



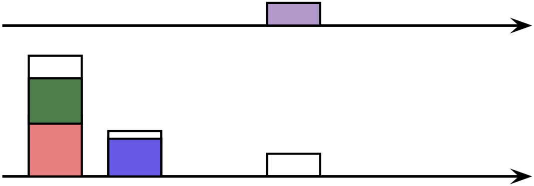
$y:$

Approximation algorithm for BPWR (2)

Obtain $y \in \{0,1\}^m$: $\rightarrow$ repair to feasible solution

- $A_i y \geq A_i x = i$ (y reserves $\geq i$ slots for largest i items)
  Buy $O(\log m) \cdot \frac{1}{s_i}$ slots for each size class $\rightarrow O(\log m \cdot \log \frac{1}{\varepsilon})$
- $|c^T x - c^T y| \leq O(1)$
- space for small items in x and y differs by $O(1)$

Input:
(large items)



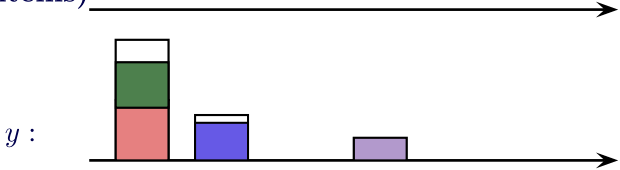
y :

Approximation algorithm for BPWR (2)

Obtain $y \in \{0,1\}^m$: $\rightarrow$ repair to feasible solution

- $A_i y \geq A_i x = i$ (y reserves $\geq i$ slots for largest i items)
  Buy $O(\log m) \cdot \frac{1}{s_i}$ slots for each size class $\rightarrow O(\log m \cdot \log \frac{1}{\varepsilon})$
- $|c^T x - c^T y| \leq O(1)$
- space for small items in x and y differs by $O(1)$

Input:
(large items)



Approximation algorithm for BPWR (2)

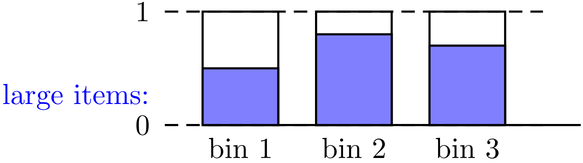
Obtain $y \in \{0,1\}^m$: $\rightarrow$ repair to feasible solution

- $A_i y \geq A_i x = i$ (y reserves $\geq i$ slots for largest i items)
  Buy $O(\log m) \cdot \frac{1}{s_i}$ slots for each size class $\rightarrow O(\log m \cdot \log \frac{1}{\varepsilon})$
- $|c^T x - c^T y| \leq O(1)$
- space for small items in x and y differs by $O(1)$

Approximation algorithm for BPWR (2)

Obtain $y \in \{0,1\}^m$: $\to$ repair to feasible solution

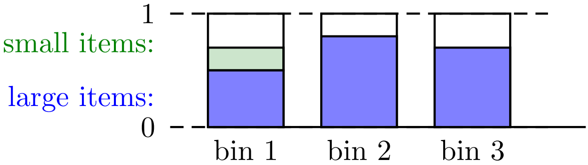
- $A_i y \geq A_i x = i$ (y reserves $\geq i$ slots for largest i items)
  Buy $O(\log m) \cdot \frac{1}{s_i}$ slots for each size class $\to O(\log m \cdot \log \frac{1}{\varepsilon})$
- $|c^T x - c^T y| \leq O(1)$
- y reserves enough space for **small** items $\to O(1)$



Approximation algorithm for BPWR (2)

Obtain $y \in \{0,1\}^m$: $\rightarrow$ repair to feasible solution

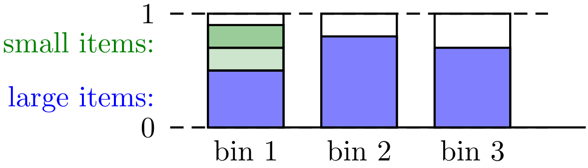
- $A_i y \geq A_i x = i$ (y reserves $\geq i$ slots for largest i items)
  Buy $O(\log m) \cdot \frac{1}{s_i}$ slots for each size class $\rightarrow O(\log m \cdot \log \frac{1}{\varepsilon})$
- $|c^T x - c^T y| \leq O(1)$
- y reserves enough space for **small** items $\rightarrow O(1)$



Approximation algorithm for BPWR (2)

Obtain $y \in \{0,1\}^m$: $\rightarrow$ repair to feasible solution

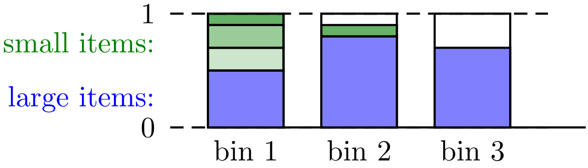
- $A_i y \geq A_i x = i$ (y reserves $\geq i$ slots for largest i items)
  Buy $O(\log m) \cdot \frac{1}{s_i}$ slots for each size class $\rightarrow O(\log m \cdot \log \frac{1}{\varepsilon})$
- $|c^T x - c^T y| \leq O(1)$
- y reserves enough space for **small** items $\rightarrow O(1)$



Approximation algorithm for BPWR (2)

Obtain $y \in \{0,1\}^m$: $\rightarrow$ repair to feasible solution

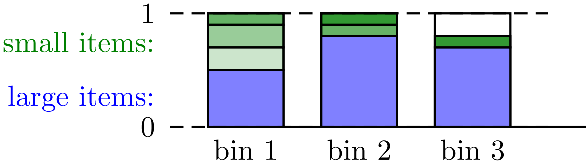
- $A_i y \geq A_i x = i$ (y reserves $\geq i$ slots for largest i items)
  Buy $O(\log m) \cdot \frac{1}{s_i}$ slots for each size class $\rightarrow O(\log m \cdot \log \frac{1}{\varepsilon})$
- $|c^T x - c^T y| \leq O(1)$
- y reserves enough space for **small** items $\rightarrow O(1)$



Approximation algorithm for BPWR (2)

Obtain $y \in \{0,1\}^m$: $\rightarrow$ repair to feasible solution

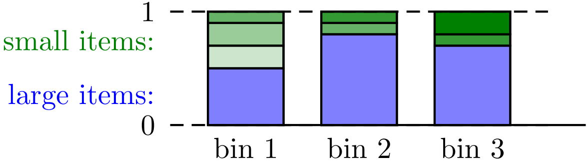
- $A_i y \geq A_i x = i$ (y reserves $\geq i$ slots for largest i items)
  Buy $O(\log m) \cdot \frac{1}{s_i}$ slots for each size class $\rightarrow O(\log m \cdot \log \frac{1}{\varepsilon})$
- $|c^T x - c^T y| \leq O(1)$
- y reserves enough space for **small** items $\rightarrow O(1)$



Approximation algorithm for BPWR (2)

Obtain $y \in \{0,1\}^m$: $\rightarrow$ repair to feasible solution

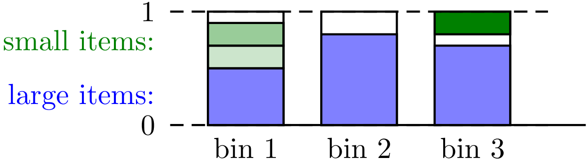
- $A_i y \geq A_i x = i$ (y reserves $\geq i$ slots for largest i items)
  Buy $O(\log m) \cdot \frac{1}{s_i}$ slots for each size class $\rightarrow O(\log m \cdot \log \frac{1}{\varepsilon})$
- $|c^T x - c^T y| \leq O(1)$
- y reserves enough space for **small** items $\rightarrow O(1)$



Approximation algorithm for BPWR (2)

Obtain $y \in \{0,1\}^m$: $\rightarrow$ repair to feasible solution

- $A_i y \geq A_i x = i$ (y reserves $\geq i$ slots for largest i items)
  Buy $O(\log m) \cdot \frac{1}{s_i}$ slots for each size class $\rightarrow O(\log m \cdot \log \frac{1}{\varepsilon})$
- $|c^T x - c^T y| \leq O(1)$
- y reserves enough space for **small** items $\rightarrow O(1)$
  assign small items integrally
  $\rightarrow O(\varepsilon) \cdot OPT_f = O(\log OPT_f)$



Approximation algorithm for BPWR (2)

Obtain $y \in \{0,1\}^m$: $\rightarrow$ repair to feasible solution

- $A_i y \geq A_i x = i$ (y reserves $\geq i$ slots for largest i items)
  Buy $O(\log m) \cdot \frac{1}{s_i}$ slots for each size class $\rightarrow O(\log m \cdot \log \frac{1}{\varepsilon})$
- $|c^T x - c^T y| \leq O(1)$
- y reserves enough space for **small** items $\rightarrow O(1)$
  assign small items integrally
  $\rightarrow O(\varepsilon) \cdot OPT_f = O(\log OPT_f)$

Doing the math:

- may assume that $x_S \leq 1 - \varepsilon$ (o.w. buy them apriori)

Approximation algorithm for BPWR (2)

Obtain $y \in \{0,1\}^m$: $\rightarrow$ repair to feasible solution

- $A_i y \geq A_i x = i$ (y reserves $\geq i$ slots for largest i items)
  Buy $O(\log m) \cdot \frac{1}{s_i}$ slots for each size class $\rightarrow O(\log m \cdot \log \frac{1}{\varepsilon})$
- $|c^T x - c^T y| \leq O(1)$
- y reserves enough space for **small** items $\rightarrow O(1)$
  assign small items integrally
  $\rightarrow O(\varepsilon) \cdot OPT_f = O(\log OPT_f)$

Doing the math:

- may assume that $x_S \leq 1 - \varepsilon$ (o.w. buy them apriori)
- Can assume that $m \leq$ #large items $+ 2$

Approximation algorithm for BPWR (2)

Obtain $y \in \{0,1\}^m$: $\rightarrow$ repair to feasible solution

- $A_i y \geq A_i x = i$ (y reserves $\geq i$ slots for largest i items)
  Buy $O(\log m) \cdot \frac{1}{s_i}$ slots for each size class $\rightarrow O(\log m \cdot \log \frac{1}{\varepsilon})$
- $|c^T x - c^T y| \leq O(1)$
- y reserves enough space for **small** items $\rightarrow O(1)$
  assign small items integrally
  $\rightarrow O(\varepsilon) \cdot OPT_f = O(\log OPT_f)$

Doing the math:

- may assume that $x_S \leq 1 - \varepsilon$ (o.w. buy them apriori)
- Can assume that $m \leq \#\text{large items} + 2$
- $OPT_f \geq \varepsilon^2 \cdot \#\text{large items} \Rightarrow \log m, \log \frac{1}{\varepsilon} = O(\log OPT_f)$

Approximation algorithm for BPWR (2)

Obtain $y \in \{0,1\}^m$: $\rightarrow$ repair to feasible solution

- $A_i y \geq A_i x = i$ (y reserves $\geq i$ slots for largest i items)
  Buy $O(\log m) \cdot \frac{1}{s_i}$ slots for each size class $\rightarrow O(\log m \cdot \log \frac{1}{\varepsilon})$
- $|c^T x - c^T y| \leq O(1)$
- y reserves enough space for **small** items $\rightarrow O(1)$
  assign small items integrally
  $\rightarrow O(\varepsilon) \cdot OPT_f = O(\log OPT_f)$

Doing the math:

- may assume that $x_S \leq 1 - \varepsilon$ (o.w. buy them apriori)
- Can assume that $m \leq$ #large items $+ 2$
- $OPT_f \geq \varepsilon^2 \cdot$ #large items $\Rightarrow \log m, \log \frac{1}{\varepsilon} = O(\log OPT_f)$
- $APX - OPT_f \leq O(\log^2 OPT_f)$ $\square$

Open problem I

Method works pretty well for other Bin Packing variants. But:

Open question I

Are there other applications?

Open problem II

Bin Packing:

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

Modified Integer Roundup Conjecture

$OPT \leq \lceil OPT_f \rceil + 1$

- True, if # of different item sizes ≤ 7 [Sebő, Shmonin '09]
- Best known general bound: $OPT \leq OPT_f + O(\log^2 n)$
- WARNING: No $o(\log^2 n)$ bound possible by just "selecting patterns from an initial fractional solution and rounding up items" [Eisenbrand, Pálvölgyi, R. '11]

Open problem II

Bin Packing:

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

Modified Integer Roundup Conjecture

$OPT \leq \lceil OPT_f \rceil + 1$

- True, if # of different item sizes ≤ 7 [Sebő, Shmonin '09]
- Best known general bound: $OPT \leq OPT_f + O(\log^2 n)$
- WARNING: No $o(\log^2 n)$ bound possible by just "selecting patterns from an initial fractional solution and rounding up items" [Eisenbrand, Pálvölgyi, R. '11]

Thanks for your attention