Non-Uniform Graph Partitioning

Robert Krauthgamer
Weizmann Institute

Seffi Naor
Technion

Roy Schwartz
Microsoft Research

Kunal Talwar
Microsoft Research

Problem Definition

Non-Uniform Graph Partitioning

Input:

- $G = (V, E)$, $w : E \to \mathcal{R}_+$.
- Capacities $n_1, n_2, \ldots, n_k$ s.t. $\sum_{j=1}^{k} n_j \geqslant n$.

Problem Definition

Non-Uniform Graph Partitioning

Input:

- $G = (V, E)$, $w : E \to \mathcal{R}_+$.
- Capacities $n_1, n_2, \ldots, n_k$ s.t. $\sum_{j=1}^{k} n_j \geqslant n$.

Output:

- A partition $S_1, \ldots, S_k$ of V where each $|S_j| \leqslant n_j$ minimizing:

$$\frac{1}{2} \sum_{j=1}^{k} \delta(S_j) \ .$$

Problem Definition

Non-Uniform Graph Partitioning

Input:

- $G = (V, E)$, $w : E \to \mathcal{R}_+$.
- Capacities $n_1, n_2, \ldots, n_k$ s.t. $\sum_{j=1}^{k} n_j \geqslant n$.

Output:

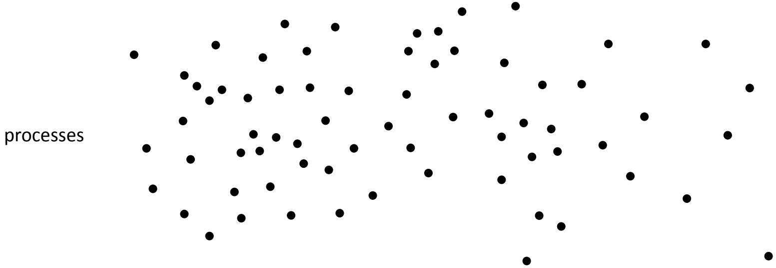
- A partition $S_1, \ldots, S_k$ of V where each $|S_j| \leqslant n_j$ minimizing:

$$\frac{1}{2} \sum_{j=1}^{k} \delta(S_j) \ .$$

Note:

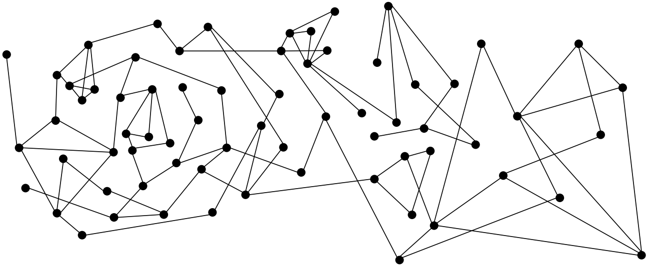
- Number of parts k might depend on n.
- Capacities might be of different magnitudes.

Example - Cloud Computing

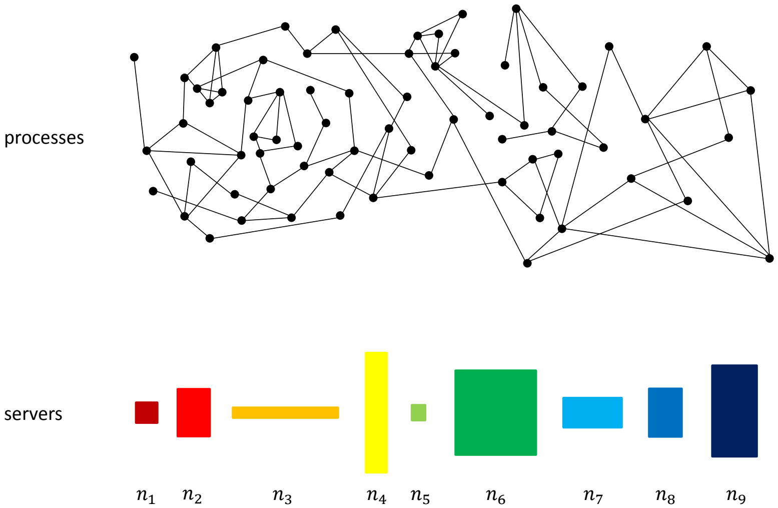


Example - Cloud Computing

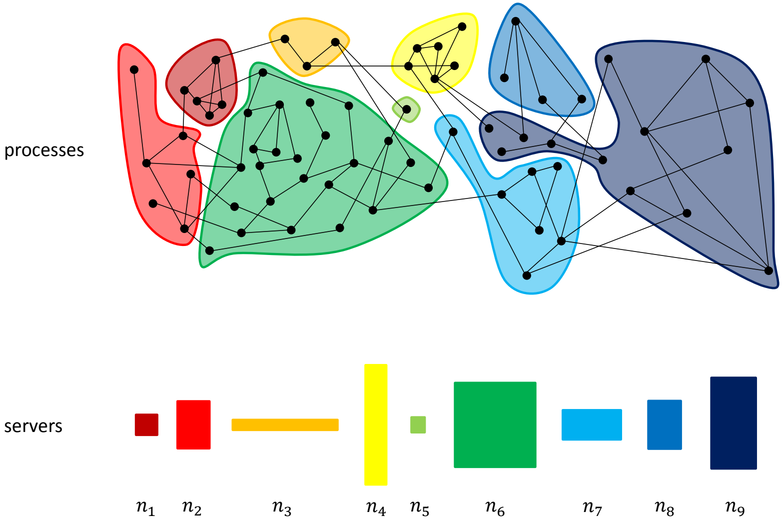
processes



Example - Cloud Computing



Example - Cloud Computing



Motivation

Theoretical:

- Captures well studied problems:
  Min-Bisection, Min b-Balanced-Cut, Min k-Partitioning.

Motivation

Theoretical:

- Captures well studied problems:
  Min-Bisection, Min b-Balanced-Cut, Min k-Partitioning.

Practical:

- **Cloud and Parallel Computing:** parallelism.
- **Hardware design:** VLSI layout, circuit testing.
- **Data mining:** clustering.
- **Social network analysis:** community discovery.
- **Vision:** pattern recognition.
- **Scientific Computing:** linear systems.

Related Work

Heuristics:

[Barnes-82], [Barnes-Vanneli-Walker-88], [Sanchis-89], [Hadley-Mark-Vanneli-92], [Rendl-Wolkowicz-95] ...

Mainly use spectral theory, local search and quadratic programming.

Related Work

Heuristics:

[Barnes-82], [Barnes-Vanneli-Walker-88], [Sanchis-89], [Hadley-Mark-Vanneli-92], [Rendl-Wolkowicz-95] ...

Mainly use spectral theory, local search and quadratic programming.

Worst Case Guarantee

No meaningful known bounds.

Related Work (Cont.)

Balanced Graph Partitioning: $(n_j \equiv {}^n\!/_k)$

Related Work (Cont.)

Balanced Graph Partitioning: ($\mathbf{n_j \equiv {}^{n}/_{k}}$)

- Min-Bisection: $(k = 2)$

True Approx. $\begin{cases} O(\log^{3/2} n) & \text{[Feige-Krauthgamer-02]} \\ O(\log n) & \text{[Räcke-08]} \end{cases}$

Related Work (Cont.)

Balanced Graph Partitioning: $(\mathbf{n_j \equiv {}^{n}/_{k}})$

- Min-Bisection: $(k = 2)$

True Approx. $\begin{cases} O(\log^{3/2} n) & \text{[Feige-Krauthgamer-02]} \\ O(\log n) & \text{[Räcke-08]} \end{cases}$

Bicriteria Approx. $\begin{cases} O(\log n) & \text{[Leighton-Rao-99]} \\ O\left(\sqrt{\log n}\right) & \text{[Arora-Rao-Vazirani-08]} \end{cases}$

Related to Sparsest-Cut and Min b-Balanced-Cut.

Related Work (Cont.)

Balanced Graph Partitioning: ($\mathbf{n_j \equiv {}^{n}/_{k}}$)

- Min-Bisection: ($k = 2$)

True Approx.	$O(\log^{3/2} n)$	[Feige-Krauthgamer-02]
	$O(\log n)$	[Räcke-08]
Bicriteria Approx.	$O(\log n)$	[Leighton-Rao-99]
	$O\left(\sqrt{\log n}\right)$	[Arora-Rao-Vazirani-08]

Related to Sparsest-Cut and Min b-Balanced-Cut.

- Min k-Partitioning: (general k)

NP-Hardness — no true approximation — [Andreev-Räcke-06]

Related Work (Cont.)

Balanced Graph Partitioning: ($\mathbf{n_j \equiv {}^{n}/_{k}}$)

- Min-Bisection: ($k = 2$)

True Approx. $\begin{cases} O(\log^{3/2} n) & \text{[Feige-Krauthgamer-02]} \\ O(\log n) & \text{[Räcke-08]} \end{cases}$

Bicriteria Approx. $\begin{cases} O(\log n) & \text{[Leighton-Rao-99]} \\ O(\sqrt{\log n}) & \text{[Arora-Rao-Vazirani-08]} \end{cases}$

Related to Sparsest-Cut and Min b-Balanced-Cut.

- Min k-Partitioning: (general k)

NP-Hardness — no true approximation — [Andreev-Räcke-06]

Bicriteria Approx.

$\begin{cases} (O(\log n), 2) & \text{[Even-Naor-Rao-Schieber-99]} \\ (O(\sqrt{\log n \log k}), 2) & \text{[Krauthgamer-Naor-S-09]} \end{cases}$

$\begin{cases} (O(\varepsilon^{-2} \log^{3/2} n), 1+\varepsilon) & \text{[Andreev-Räcke-06]} \\ (O(\log n), 1+\varepsilon) & \text{[Feldmann-Forschini-12]} \end{cases}$

Related Work (Cont.)

Capacitated Metric Labeling:

- The same as Non-Uniform Graph Partitioning with additional assignment costs.

$(O(\log n), O((\log k)))$	$n_j \equiv n/k$	[Naor-S-05]
$(O(\log n), 1)$	constant k	[Andrews-Hajiaghayi-Karloff-Moitra-11]

- $NP \not\subseteq ZPTIME\left(n^{polylog(n)}\right) \implies$
  No finite approximation that violates capacities by $O(\log^{1/2-\varepsilon} k)$, $\forall \varepsilon > 0$.
  [Andrews-Hajiaghayi-Karloff-Moitra-11]

Our Result

Theorem [Krauthgamer-Naor-S-Talwar-13]

There is a bicriteria approximation algorithm achieving a guarantee of:

$$(O(\log n), O(1))$$

for Non-Uniform Graph Partitioning.

Known Techniques - Issues

1. Recursive Partitioning:
   How many vertices to cut in each step?

Known Techniques - Issues

1. Recursive Partitioning:
   How many vertices to cut in each step?
2. Spreading Metrics:
   Only spreading with average capacity - insufficient.

Known Techniques - Issues

1. Recursive Partitioning:
   How many vertices to cut in each step?
2. Spreading Metrics:
   Only spreading with average capacity - insufficient.
3. Räcke's Tree Decomposition:
   Dynamic programming does not seem to yield poly running time.

Known Techniques - Issues

1. Recursive Partitioning:
   How many vertices to cut in each step?
2. Spreading Metrics:
   Only spreading with average capacity - insufficient.
3. Räcke's Tree Decomposition:
   Dynamic programming does not seem to yield poly running time.

Our Approach

- Configuration LP.
- Randomized rounding $+$ concentration via stopping times.

Configuration LP

$$\mathcal{F}_j \triangleq \{S \ : \ S \subseteq V, |S| \leqslant n_j\}$$

Configuration LP

$$\mathcal{F}_j \triangleq \{S \ : \ S \subseteq V, |S| \leqslant n_j\}$$

$$\begin{aligned}
(\mathcal{P}) \quad \min \quad & \frac{1}{2}\sum_{j=1}^{k} \sum_{S \in \mathcal{F}_j} \delta(S) \cdot x_{S,j} & \\
s.t. \quad & \sum_{j=1}^{k} \sum_{S \in \mathcal{F}_j : u \in S} x_{S,j} \geqslant 1 & \forall u \in V \\
& \sum_{S \in \mathcal{F}_j} x_{S,j} \leqslant 1 & \forall j = 1, \ldots, k \\
& x_{S,j} \geqslant 0 & \forall j = 1, \ldots, k, \forall S \in \mathcal{F}_j
\end{aligned}$$

Configuration LP (Cont.)

Theorem

$(\mathcal{P})$ can be efficiently solved up to a loss of $O(\log n)$ in the objective.

Configuration LP (Cont.)

Theorem

$(\mathcal{P})$ can be efficiently solved up to a loss of $O(\log n)$ in the objective.

Proof Outline:

- Dual separation oracle of $(\mathcal{P})$ relates to Min ρ-Unbalanced-Cut. Techniques from [Räcke-08] give an $O(\log n)$ approximation.

Configuration LP (Cont.)

Theorem

$(\mathcal{P})$ can be efficiently solved up to a loss of $O(\log n)$ in the objective.

Proof Outline:

- Dual separation oracle of $(\mathcal{P})$ relates to Min ρ-Unbalanced-Cut. Techniques from [Räcke-08] give an $O(\log n)$ approximation.
- Difficulty: Applying the above in algorithms for solving mixed packing/covering LPs yields $O(\log n)$ loss in cost and capacity.

Configuration LP (Cont.)

Theorem

$(\mathcal{P})$ can be efficiently solved up to a loss of $O(\log n)$ in the objective.

Proof Outline:

- Dual separation oracle of $(\mathcal{P})$ relates to Min ρ-Unbalanced-Cut. Techniques from [Räcke-08] give an $O(\log n)$ approximation.
- Difficulty: Applying the above in algorithms for solving mixed packing/covering LPs yields $O(\log n)$ loss in cost and capacity.

  Solution: Scaling constraints differently. ■

Randomized Rounding

Assumption:

- $n_1 \geqslant n_2 \geqslant \ldots \geqslant n_k$ and each n_j is a power of 2.

Randomized Rounding

Assumption:

- $n_1 \geqslant n_2 \geqslant \ldots \geqslant n_k$ and each n_j is a power of 2.

Notation:

$$\underbrace{n_1 = n_2 = n_3}_{W_1} > \underbrace{n_4 = n_5}_{W_2} > \underbrace{n_6 = n_7 = n_8}_{W_3} > \ldots\ldots\ldots\ldots$$

Randomized Rounding

Assumption:

- $n_1 \geqslant n_2 \geqslant \ldots \geqslant n_k$ and each n_j is a power of 2.

Notation:

$$\underbrace{n_1 = n_2 = n_3}_{W_1} > \underbrace{n_4 = n_5}_{W_2} > \underbrace{n_6 = n_7 = n_8}_{W_3} > \ldots\ldots\ldots\ldots$$

$$W_i \triangleq \left\{ j \ : \ n_j = 2^{-(i-1)} n_1 \right\} \qquad i = 1, 2, \ldots, \ell$$

$$k_i \triangleq |W_i|$$

Randomized Rounding (Cont.)

Idea: Covering vertices by random cuts.

Randomized Rounding (Cont.)

Idea: Covering vertices by random cuts.

Rounding - General Approach

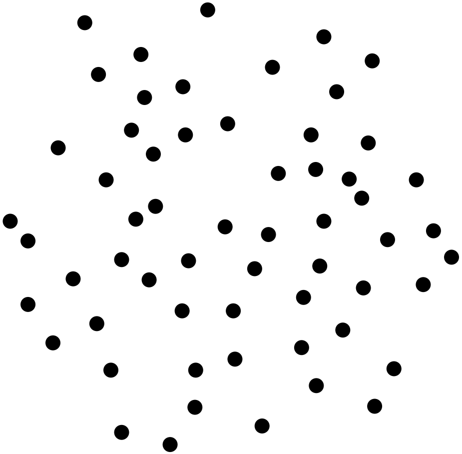
1. $H_i \leftarrow \emptyset$ for every $i = 1, \ldots, \ell$.
2. While $V \neq \emptyset$:
    - Choose $j \sim Unif[1, \ldots, k]$.
    - Choose $S \in \mathcal{F}_j$ w.p. $x_{S,j}$.
    - Let r be the mega-bucket s.t. $j \in W_r$.
    - $H_r \leftarrow H_r \cup \{S \cap V\}$.
    - $V \leftarrow V \setminus S$.

Randomized Rounding - Example

- $H_1 = \phi$
- $H_2 = \phi$
- $H_3 = \phi$

$\vdots$

- $H_\ell = \phi$

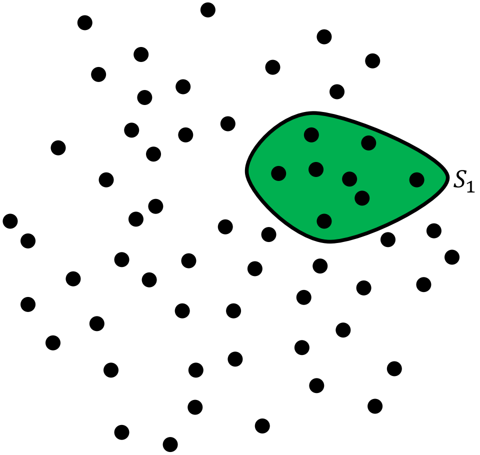


Randomized Rounding - Example

- $H_1 = \phi$
- $H_2 = \phi$
- $H_3 = \phi$

⋮

- $H_\ell = \phi$



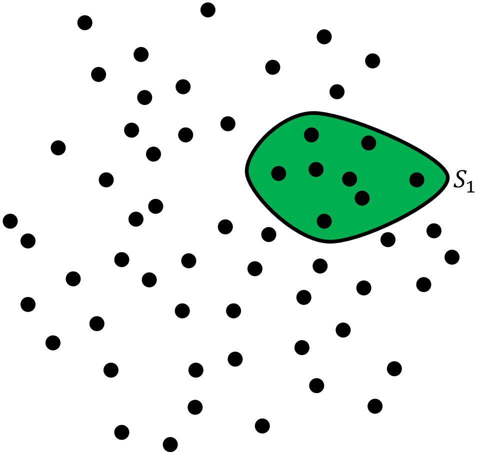
(S_1, j_1) $\quad j_1 \in W_3$

Randomized Rounding - Example

- $H_1 = \phi$
- $H_2 = \phi$
- $H_3 = \{S_1\}$

⋮

- $H_\ell = \phi$



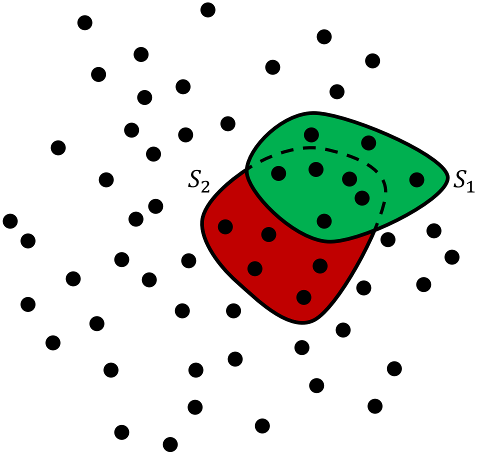
(S_1, j_1) $\quad$ $j_1 \in W_3$

Randomized Rounding - Example

- $H_1 = \phi$
- $H_2 = \phi$
- $H_3 = \{S_1\}$

⋮

- $H_\ell = \phi$



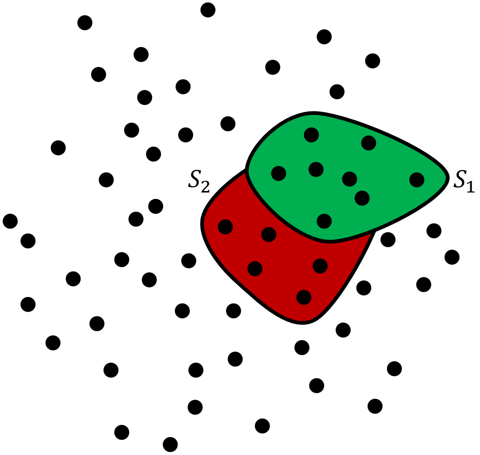
(S_2, j_2) $\quad j_2 \in W_1$

Randomized Rounding - Example

- $H_1 = \{S_2 \backslash S_1\}$
- $H_2 = \phi$
- $H_3 = \{S_1\}$

⋮

- $H_\ell = \phi$



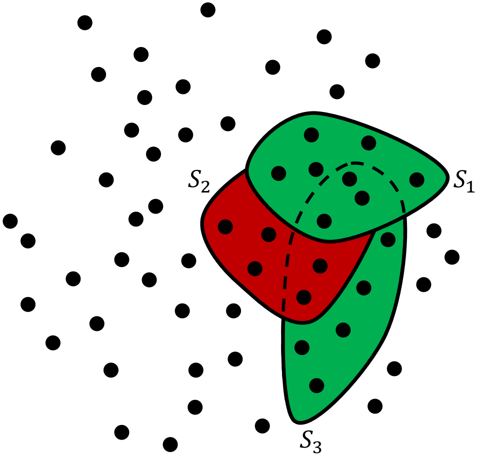
(S_2, j_2) $\quad j_2 \in W_1$

Randomized Rounding - Example

- $H_1 = \{S_2 \backslash S_1\}$
- $H_2 = \phi$
- $H_3 = \{S_1\}$

⋮

- $H_\ell = \phi$



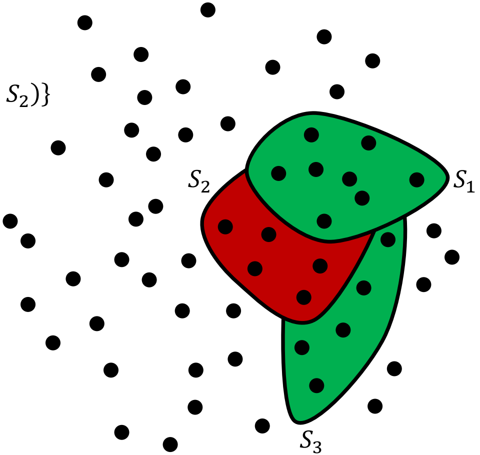
(S_3, j_3) $\quad j_3 \in W_3$

Randomized Rounding - Example

- $H_1 = \{S_2 \backslash S_1\}$
- $H_2 = \phi$
- $H_3 = \{S_1, S_3 \backslash (S_1 \cup S_2)\}$

$\vdots$

- $H_\ell = \phi$



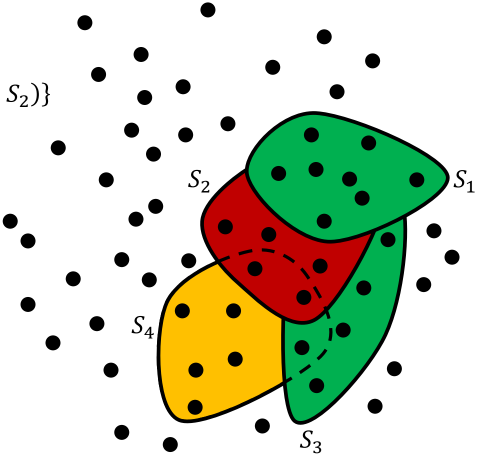
(S_3, j_3) $\quad j_3 \in W_3$

Randomized Rounding - Example

- $H_1 = \{S_2 \backslash S_1\}$
- $H_2 = \phi$
- $H_3 = \{S_1, S_3 \backslash (S_1 \cup S_2)\}$

⋮

- $H_\ell = \phi$



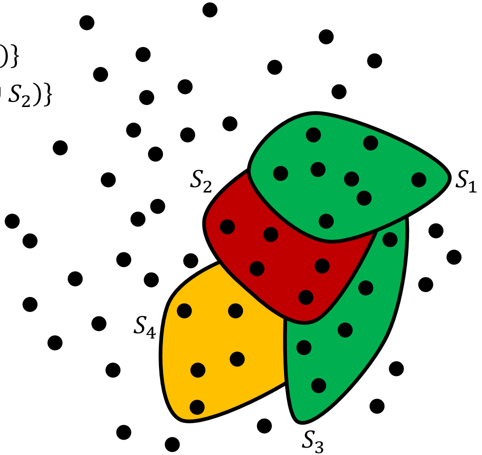
(S_4, j_4) $\quad j_4 \in W_2$

Randomized Rounding - Example

- $H_1 = \{S_2 \backslash S_1\}$
- $H_2 = \{S_4 \backslash (S_2 \cup S_3)\}$
- $H_3 = \{S_1, S_3 \backslash (S_1 \cup S_2)\}$

$\vdots$

- $H_\ell = \phi$



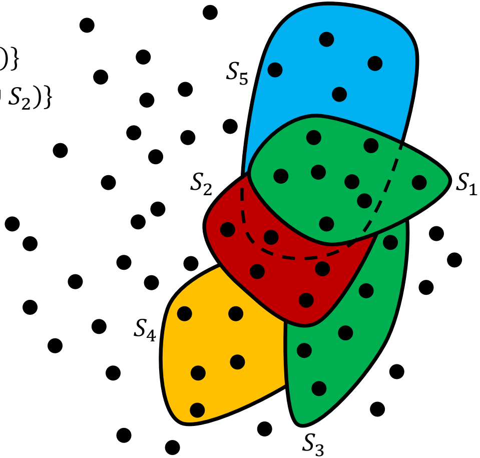
(S_4, j_4) $j_4 \in W_2$

Randomized Rounding - Example

- $H_1 = \{S_2 \backslash S_1\}$
- $H_2 = \{S_4 \backslash (S_2 \cup S_3)\}$
- $H_3 = \{S_1, S_3 \backslash (S_1 \cup S_2)\}$

$\vdots$

- $H_\ell = \phi$



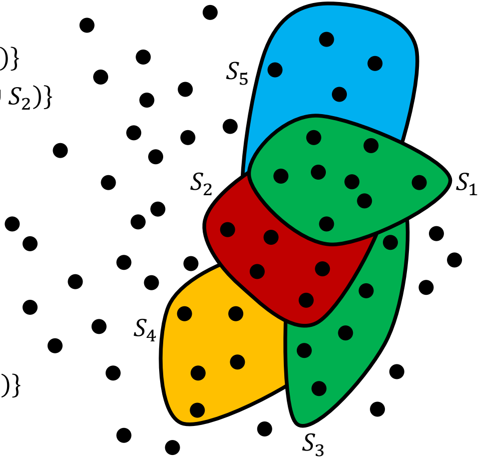
$(S_5, j_5) \qquad j_5 \in W_\ell$

Randomized Rounding - Example

- $H_1 = \{S_2 \backslash S_1\}$
- $H_2 = \{S_4 \backslash (S_2 \cup S_3)\}$
- $H_3 = \{S_1, S_3 \backslash (S_1 \cup S_2)\}$

⋮

- $H_\ell = \{S_5 \backslash (S_1 \cup S_2)\}$



(S_5, j_5) $\quad j_5 \in W_\ell$

Randomized Rounding (Cont.)

Question: What to do when all vertices are covered?

Randomized Rounding (Cont.)

Question: What to do when all vertices are covered?

Merge: While $|H_i| > k_i$ merge smallest two cuts in H_i.

Randomized Rounding (Cont.)

Question: What to do when all vertices are covered?

Merge: While $|H_i| > k_i$ merge smallest two cuts in H_i.

Theorem - Cost Analysis

$$\mathbb{E}\left[cost(ALG)\right] \leqslant 2 \cdot cost\left(\mathcal{P}\right) \ .$$

Proof Outline:

$$\Pr\left[u \text{ and } v \text{ covered in different iterations}\right] \leqslant \sum_{j=1}^{k} \sum_{S \in \mathcal{F}_j : (u,v) \in \delta(S)} x_{S,j} \ .$$

∎

Capacity Analysis - Attempt I

Observation - Interchanging Cuts Within W_i

It suffices to upper bound:

$$N_i \triangleq \text{ number of vertices covered by } H_i \text{ at the end }.$$

Capacity Analysis - Attempt I

Observation - Interchanging Cuts Within W_i

It suffices to upper bound:

$$N_i \triangleq \text{ number of vertices covered by } H_i \text{ at the end }.$$

Note:

- $\mathbb{E}\left[N_i\right] \leqslant k_i \cdot 2^{-(i-1)} n_1$.

Capacity Analysis - Attempt I

Observation - Interchanging Cuts Within W_i

It suffices to upper bound:

$$N_i \triangleq \text{ number of vertices covered by } H_i \text{ at the end }.$$

Note:

- $\mathbb{E}[N_i] \leqslant k_i \cdot 2^{-(i-1)} n_1$.
- $\ell = O(\log k)$ by merging every W_i with $k_i \leqslant 2^{1/2(i-1)}$ into W_1.
  (ℓ is number of mega-buckets)

Capacity Analysis - Attempt I

Observation - Interchanging Cuts Within W_i

It suffices to upper bound:

$$N_i \triangleq \text{ number of vertices covered by } H_i \text{ at the end .}$$

Note:

- $\mathbb{E}\left[N_i\right] \leqslant k_i \cdot 2^{-(i-1)} n_1$.
- $\ell = O(\log k)$ by merging every W_i with $k_i \leqslant 2^{1/2(i-1)}$ into W_1.
  (ℓ is number of mega-buckets)

Conclusion: Markov $+$ union bound $\Rightarrow$ $O(\log k)$ capacity violation.

Capacity Analysis - Attempt II

Martingale

$$M_{i,t} \triangleq \mathbb{E}\left[N_i \mid (S_1, j_1), \ldots, (S_t, j_t)\right]$$

$\{M_{i,t}\}_{t=0}^{\infty}$ is a martingale with respect to $\{(S_t, j_t)\}_{t=1}^{\infty}$.

Capacity Analysis - Attempt II

Martingale

$$M_{i,t} \triangleq \mathbb{E}\left[N_i \mid (S_1, j_1), \ldots, (S_t, j_t)\right]$$

$\{M_{i,t}\}_{t=0}^{\infty}$ is a martingale with respect to $\{(S_t, j_t)\}_{t=1}^{\infty}$.

Note:

- $M_{i,0} = \mathbb{E}\left[N_i\right]$.

Capacity Analysis - Attempt II

Martingale

$$M_{i,t} \triangleq \mathbb{E}\left[N_i \mid (S_1, j_1), \ldots, (S_t, j_t)\right]$$

$\{M_{i,t}\}_{t=0}^{\infty}$ is a martingale with respect to $\{(S_t, j_t)\}_{t=1}^{\infty}$.

Note:

- $M_{i,0} = \mathbb{E}\left[N_i\right]$.
- $|M_{i,t} - M_{i,t-1}| \leqslant k_i \cdot 2^{-(i-1)} n_1$. (Lipschitz)

Capacity Analysis - Attempt II

Martingale

$$M_{i,t} \triangleq \mathbb{E}\left[N_i \mid (S_1, j_1), \ldots, (S_t, j_t)\right]$$

$\{M_{i,t}\}_{t=0}^{\infty}$ is a martingale with respect to $\{(S_t, j_t)\}_{t=1}^{\infty}$.

Note:

- $M_{i,0} = \mathbb{E}\left[N_i\right]$.
- $|M_{i,t} - M_{i,t-1}| \leqslant k_i \cdot 2^{-(i-1)} n_1$. (Lipschitz)
- Number of iterations to cover V: $T = \Theta(k \log n)$.

Capacity Analysis - Attempt II

Martingale

$$M_{i,t} \triangleq \mathbb{E}\left[N_i \mid (S_1, j_1), \ldots, (S_t, j_t)\right]$$

$\{M_{i,t}\}_{t=0}^{\infty}$ is a martingale with respect to $\{(S_t, j_t)\}_{t=1}^{\infty}$.

Note:

- $M_{i,0} = \mathbb{E}\left[N_i\right]$.
- $|M_{i,t} - M_{i,t-1}| \leqslant k_i \cdot 2^{-(i-1)} n_1$. (Lipschitz)
- Number of iterations to cover V: $T = \Theta(k \log n)$.

Conclusion: Azuma + union bound $\Rightarrow$ $O(\sqrt{k \log n \log \log k})$ capacity violation.

Capacity Analysis - Attempt III

Worst Conditional Variance:

Let $(T_1, r_1), \ldots, (T_{t-1}, r_{t-1})$ be the realization that maximizes:

$$\text{Var}\left[M_{i,t} - M_{i,t-1} | (S_1, j_1) = (T_1, r_1), \ldots, (S_{t-1}, j_{t-1}) = (T_{t-1}, r_{t-1})\right] .$$

$v_{i,t}$ is the worst variance value.

Capacity Analysis - Attempt III

Worst Conditional Variance:

Let $(T_1, r_1), \ldots, (T_{t-1}, r_{t-1})$ be the realization that maximizes:

$$\text{Var}\left[M_{i,t} - M_{i,t-1} | (S_1, j_1) = (T_1, r_1), \ldots, (S_{t-1}, j_{t-1}) = (T_{t-1}, r_{t-1})\right] .$$

$v_{i,t}$ is the worst variance value.

Note:

- For every t a different conditioning might be chosen.
- Martingale concentration via bounded variances (Bernstein) is not sufficient:

  $\sum_{t \geqslant 1} v_{i,t}$ might be too big!

Capacity Analysis - Attempt III (Cont.)

$$\Pr\left[\sum_{t\geqslant 1} \text{Var}\left[M_{i,t} - M_{i,t-1} | (S_1, j_1), \ldots, (S_{t-1}, j_{t-1})\right] \text{ is small}\right] \geqslant \text{ ?}$$

Capacity Analysis - Attempt III (Cont.)

$$\Pr\left[\sum_{t\geqslant 1}\mathsf{Var}\big[M_{i,t}-M_{i,t-1}|(S_1,j_1),\ldots,(S_{t-1},j_{t-1})\big]\ \text{is small}\right]\geqslant\ ?$$

Theorem - Conditional Variances Sum

$$\Pr\left[\sum_{t\geqslant 1}\mathsf{Var}\big[M_{i,t}-M_{i,t-1}\mid(S_1,j_1),\ldots,(S_{t-1},j_{t-1})\big]\geqslant 2\alpha\cdot k_i\cdot 2^{-(i-1)}n_1^2\right]\leqslant {}^1\!/_\alpha\ .$$

Capacity Analysis - Attempt III (Cont.)

$$\Pr\left[\sum_{t\geqslant 1} \mathsf{Var}\left[M_{i,t} - M_{i,t-1} | (S_1, j_1), \ldots, (S_{t-1}, j_{t-1})\right] \text{ is small}\right] \geqslant \text{ ?}$$

Theorem - Conditional Variances Sum

$$\Pr\left[\sum_{t\geqslant 1} \mathsf{Var}\left[M_{i,t} - M_{i,t-1} \mid (S_1, j_1), \ldots, (S_{t-1}, j_{t-1})\right] \geqslant 2\alpha \cdot k_i \cdot 2^{-(i-1)} n_1^2\right] \leqslant {}^1\!/\!_\alpha \ .$$

Intuition: In later iterations the changes are smaller in expectation. ■

Capacity Analysis - Attempt III (Cont.)

Martingale

$$\{M_{i,t}\}_{t=0}^{\infty}$$

Good Event

Variances are small.

Capacity Analysis - Attempt III (Cont.)

Martingale

$$\{M_{i,t}\}_{t=0}^{\infty}$$

Good Event

Variances are small.

$\searrow \quad \swarrow$

???

Capacity Analysis - Attempt III (Cont.)

Martingale

$$\{M_{i,t}\}_{t=0}^{\infty}$$

Good Event

Variances are small.

$\searrow \quad \swarrow$

???

Freedman's Inequality

(stopping-time based concentration)

Capacity Analysis - Attempt III (Cont.)

Immediate Conclusion:
Freedman's inequality yields $O(\log \log k)$ capacity violation.

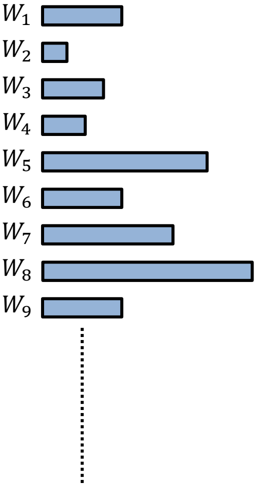
Capacity Analysis - Attempt III (Cont.)

Immediate Conclusion:
Freedman's inequality yields $O(\log \log k)$ capacity violation.

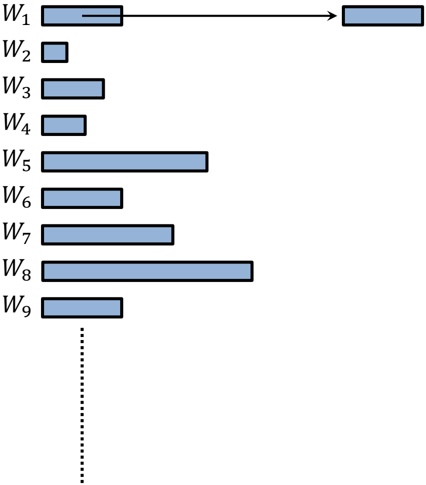
Question

How do we get $O(1)$ capacity violation?

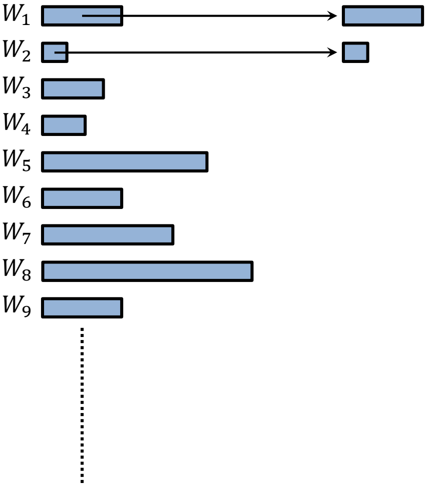
Instance Transformation



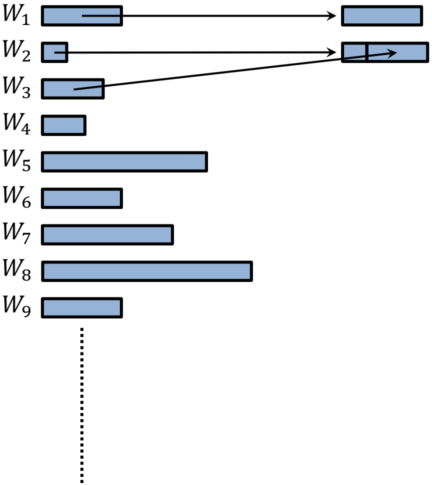
Instance Transformation



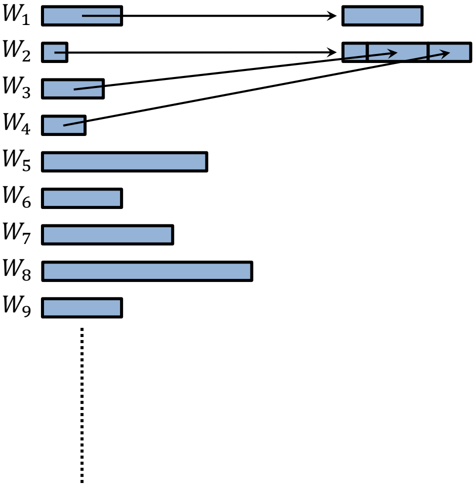
Instance Transformation



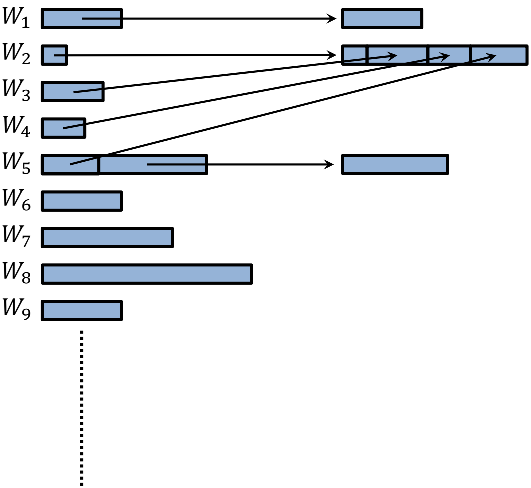
Instance Transformation



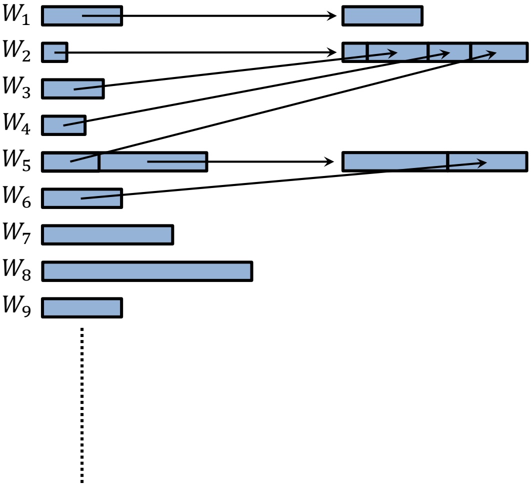
Instance Transformation



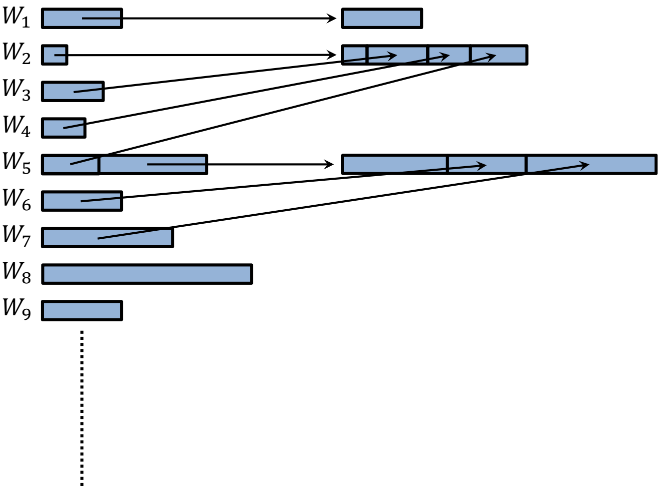
Instance Transformation



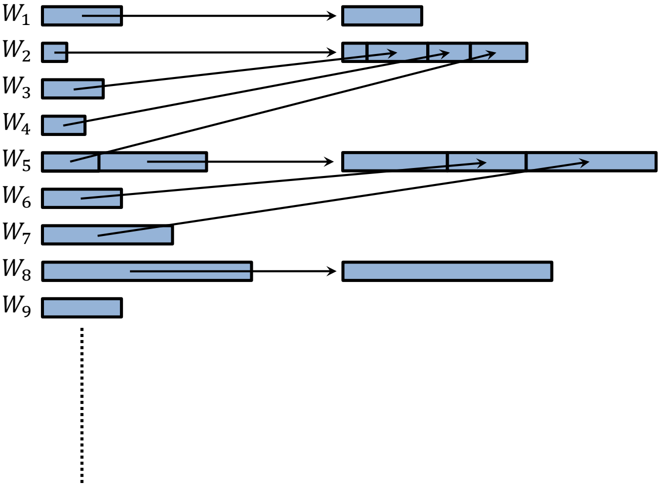
Instance Transformation



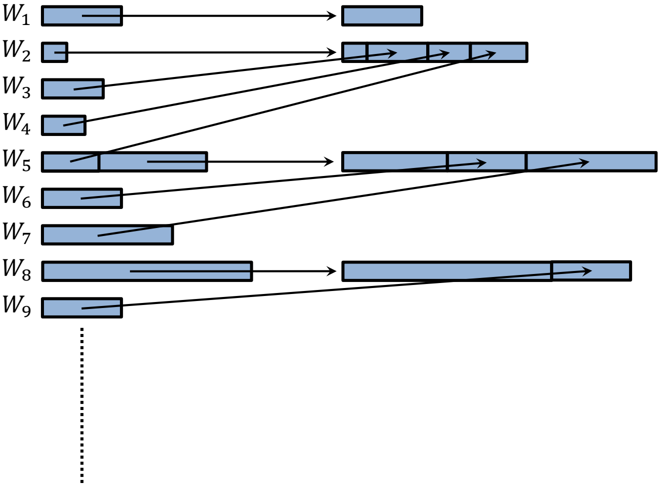
Instance Transformation



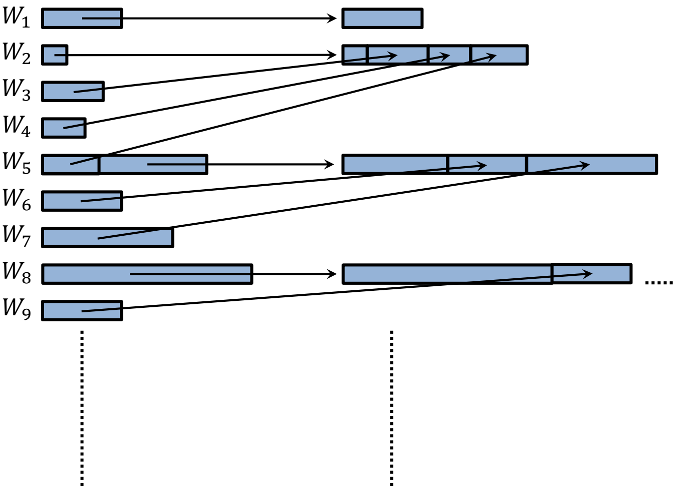
Instance Transformation



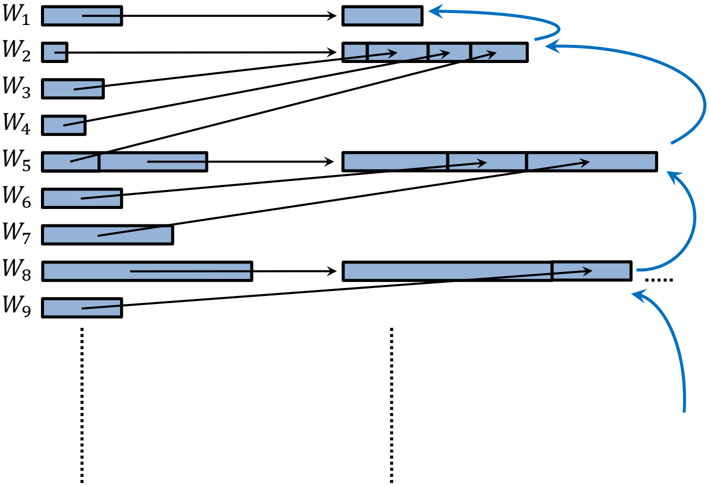
Instance Transformation



Instance Transformation



Instance Transformation



Instance Transformation (Cont.)

What is an instance transformation?

Instance Transformation (Cont.)

What is an instance transformation?

- Total capacity of non-empty W_i grows by a constant c.
- Inverse transformation moves cuts from W_i to some W_j, $j \leqslant i$.
- Inverse transformation incurs a c capacity violation.

Instance Transformation (Cont.)

What is an instance transformation?

- Total capacity of non-empty W_i grows by a constant c.
- Inverse transformation moves cuts from W_i to some W_j, $j \leqslant i$.
- Inverse transformation incurs a c capacity violation.

What does an instance transformation provide?

Instance Transformation (Cont.)

What is an instance transformation?

- Total capacity of non-empty W_i grows by a constant c.
- Inverse transformation moves cuts from W_i to some W_j, $j \leqslant i$.
- Inverse transformation incurs a c capacity violation.

What does an instance transformation provide?

Large total capacity of W_i

$$\Downarrow$$

Better concentration as the total capacity increases

Instance Transformation (Cont.)

What is an instance transformation?

- Total capacity of non-empty W_i grows by a constant c.
- Inverse transformation moves cuts from W_i to some W_j, $j \leqslant i$.
- Inverse transformation incurs a c capacity violation.

What does an instance transformation provide?

Large total capacity of W_i

$\Downarrow$

Better concentration as the total capacity increases

$\Downarrow$

Sum of conditional variances might be higher

Instance Transformation (Cont.)

What is an instance transformation?

- Total capacity of non-empty W_i grows by a constant c.
- Inverse transformation moves cuts from W_i to some W_j, $j \leqslant i$.
- Inverse transformation incurs a c capacity violation.

What does an instance transformation provide?

Large total capacity of W_i

$\Downarrow$

Better concentration as the total capacity increases

$\Downarrow$

Sum of conditional variances might be higher

$\Downarrow$

Probability that sum of conditional variances is small increases

Instance Transformation (Cont.)

What is an instance transformation?

- Total capacity of non-empty W_i grows by a constant c.
- Inverse transformation moves cuts from W_i to some W_j, $j \leqslant i$.
- Inverse transformation incurs a c capacity violation.

What does an instance transformation provide?

Large total capacity of W_i

$\Downarrow$

Better concentration as the total capacity increases

$\Downarrow$

Sum of conditional variances might be higher

$\Downarrow$

Probability that sum of conditional variances is small increases

Conclusion: $O(1)$ capacity violation.

Thank You!

Questions?