Algorithm Theory

Chapter 8
Approximation Algorithms

Part III:
Minimum Set Cover

Fabian Kuhn

Set Cover

Input:

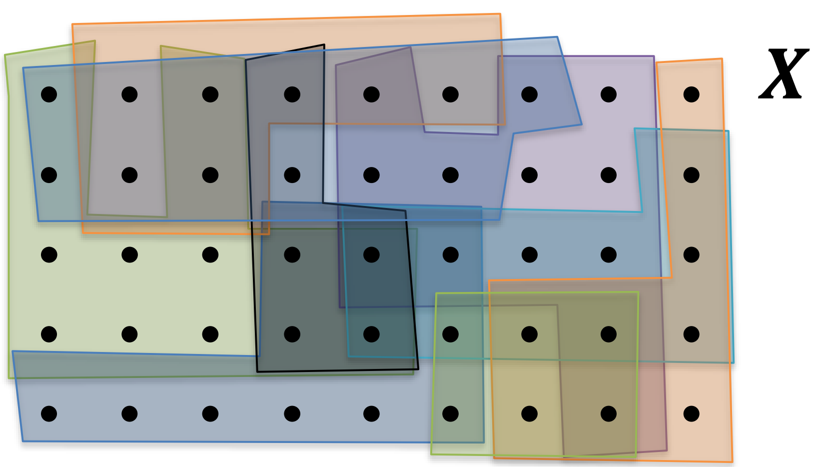
- A set of elements X and a collection $\mathcal{S}$ of subsets X, i.e., $\mathcal{S} \subseteq 2^X$
  - such that $\bigcup_{S \in \mathcal{S}} S = X$

Set Cover:

- A set cover $\mathcal{C}$ of $(X, \mathcal{S})$ is a subset of the sets $\mathcal{S}$ which covers X:

$$\bigcup_{S \in \mathcal{C}} S = X$$

Example:



Minimum (Weighted) Set Cover

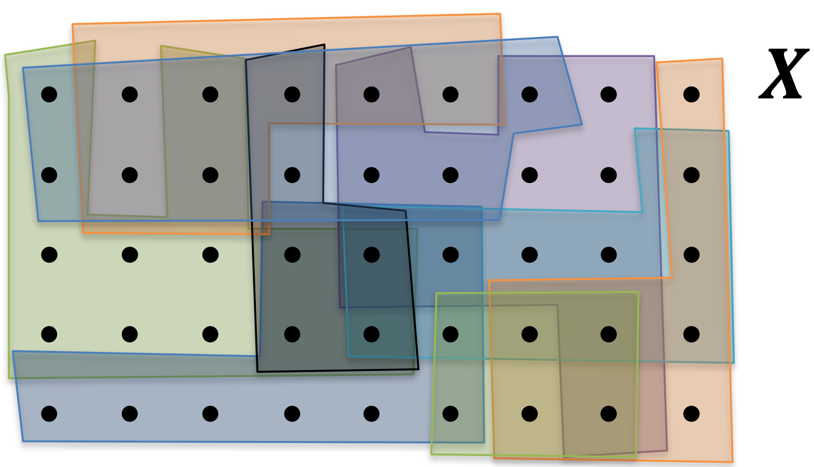
Minimum Set Cover:

- **Goal:** Find a set cover $\mathcal{C}$ of smallest possible size
  - i.e., over X with as few sets as possible

Minimum Weighted Set Cover:

- Each set $S \in \mathcal{S}$ has a weight $w_S > 0$
- **Goal:** Find a set cover $\mathcal{C}$ of minimum weight

Example:

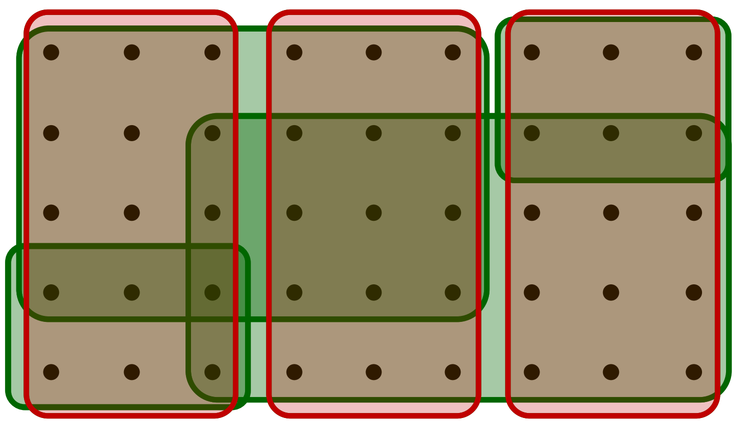


Minimum Set Cover: Greedy Algorithm

Greedy Set Cover Algorithm:

- Start with $\mathcal{C} = \emptyset$
- In each step, add set $S \in \mathcal{S} \setminus \mathcal{C}$ to $\mathcal{C}$ s.t. S covers as many uncovered elements as possible

Example:



Weighted Set Cover: Greedy Algorithm

Greedy Weighted Set Cover Algorithm:

- Start with $\mathcal{C} = \emptyset$
- In each step, add set $S \in \mathcal{S} \setminus \mathcal{C}$ with the best weight per newly covered element ratio (set with best efficiency):

$$S = \arg\min_{S \in \mathcal{S} \setminus \mathcal{C}} \frac{w_S}{\left|S \setminus \bigcup_{T \in \mathcal{C}} T\right|}$$

Analysis of Greedy Algorithm:

- Assign a price $p(x)$ to each element $x \in X$:
  The efficiency of the set when covering the element
- If covering x with set S, if partial cover is $\mathcal{C}$ before adding S to $\mathcal{C}$:

$$p(e) = \frac{w_S}{\left|S \setminus \bigcup_{T \in \mathcal{C}} T\right|}$$

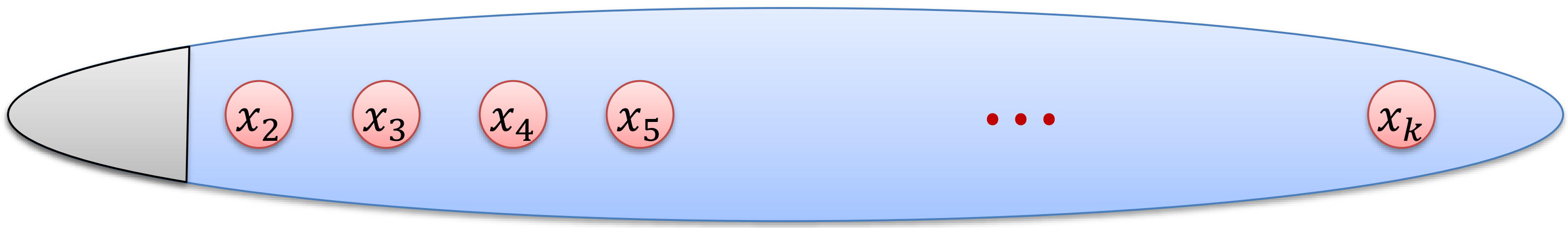
At all times:

$$\sum_{x \in X} p(x) = \sum_{S \in \mathcal{C}} w_S$$

Weighted Set Cover: Greedy Algorithm

Lemma: Consider a set $S = \{x_1, x_2, \dots, x_k\} \in \mathcal{S}$ be a set and assume that the elements are covered in the order $x_1, x_2, \dots, x_k$ by the greedy algorithm (ties broken arbitrarily).

Then, the price of element x_i is at most $p(x_i) \leq \frac{w_S}{k-i+1}$

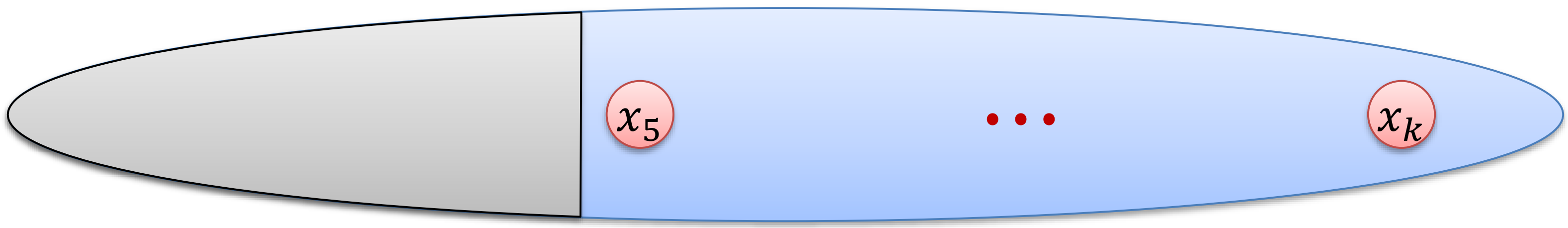


- Price of x_1 : $p(x_1) \leq \frac{w_S}{k}$
  - When x_1 gets covered, all k elments of S are uncovered
  - We therefore take a set with weight per newly covered element $\leq w_S/k$
- Price of x_2 : $p(x_2) \leq \frac{w_S}{k-1}$
  - When x_2 gets covered, $\geq k-1$ elements of S are still uncovered
  - We therefore take a set with weight per newly cov. elem. $\leq w_S/(k-1)$

Weighted Set Cover: Greedy Algorithm

Lemma: Consider a set $S = \{x_1, x_2, \dots, x_k\} \in \mathcal{S}$ be a set and assume that the elements are covered in the order $x_1, x_2, \dots, x_k$ by the greedy algorithm (ties broken arbitrarily).

Then, the price of element x_i is at most $p(x_i) \leq \frac{w_S}{k-i+1}$



- Price of x_i : $p(x_i) \leq \frac{w_S}{k-i+1}$
  - When x_i gets covered, all elements $x_i, x_{i+1}, \dots, x_k$ are still uncovered
  - We therefore take a set with weight per newly covered element

$$\leq \frac{w_S}{k-(i-1)} = \frac{w_S}{k-i+1}$$

Weighted Set Cover: Greedy Algorithm

Lemma: Consider a set $S = \{x_1, x_2, \dots, x_k\} \in \mathcal{S}$ be a set and assume that the elements are covered in the order $x_1, x_2, \dots, x_k$ by the greedy algorithm (ties broken arbitrarily).

Then, the price of element x_i is at most $p(x_i) \leq \frac{w_S}{k-i+1}$

Corollary: The total price of a set $S \in \mathcal{S}$ of size $|S| = k$ is

$$\sum_{x \in S} p(x) \leq w_S \cdot H_k, \qquad \text{where } H_k = \sum_{i=1}^{k} \frac{1}{i} \leq 1 + \ln k$$

Proof:

$$\sum_{x \in S} p(x) = \sum_{i=1}^{k} p(x_i) \leq w_S \cdot \sum_{i=1}^{k} \frac{1}{k-i+1} = w_S \cdot \sum_{j=1}^{k} \frac{1}{j}$$

Weighted Set Cover: Greedy Algorithm

Corollary: The total price of a set $S \in \mathcal{S}$ of size $|S| = k$ is

$$\sum_{x \in S} p(x) \leq w_S \cdot H_k, \qquad \text{where } H_k = \sum_{i=1}^{k} \frac{1}{i} \leq 1 + \ln k$$

Theorem: The approximation ratio of the greedy minimum (weighted) set cover algorithm is at most $\boldsymbol{H_K \leq 1 + \ln K}$, where s is the cardinality of the largest set ($K = \max_{S \in \mathcal{S}} |S|$).

- Consider the greedy solution $\mathcal{C}$ and an optimal solution $\mathcal{C}^*$:

$$w(\mathcal{C}) = \sum_{x \in X} p(x) \leq \sum_{S \in \mathcal{C}^*} \sum_{x \in S} p(x) \leq \sum_{S \in \mathcal{C}^*} w_S \cdot H_{|S|} \leq H_K \cdot w(\mathcal{C}^*)$$

$\mathcal{C}$: greedy solution

$$w(\mathcal{C}) \coloneqq \sum_{S \in \mathcal{C}} w_S$$

$\mathcal{C}^*$: optimal solution

$$w(\mathcal{C}^*) \coloneqq \sum_{S \in \mathcal{C}^*} w_S$$

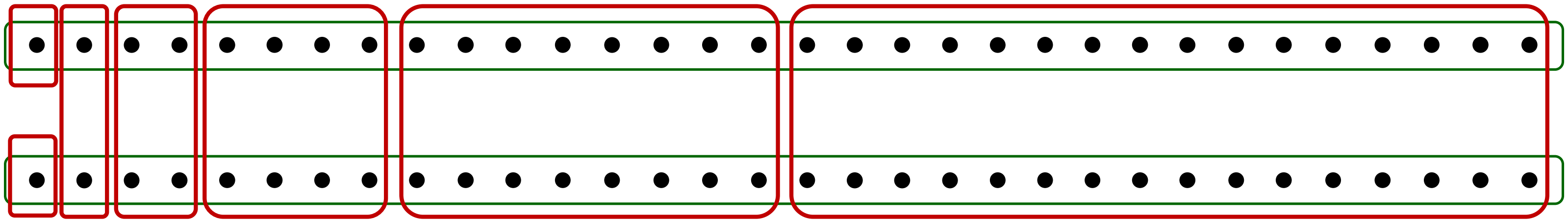
Set Cover Greedy Algorithm

Can we improve this analysis?

No! Even for the unweighted minimum set cover problem, the approximation ratio of the greedy algorithm is $\geq \big(1 - o(1)\big) \cdot \ln s$.

- if s is the size of the largest set... (s can be linear in n)

Let's show that the approximation ratio is at least $\Omega(\log n)$...



$\mathbf{OPT = 2}$

$\mathbf{GREEDY \geq \log_2 n}$

Set Cover: Better Algorithm?

An approximation ratio of $\ln n$ seems not spectacular...

Can we improve the approximation ratio?

No, unfortunately not, unless $\mathrm{P} = \mathrm{NP}$

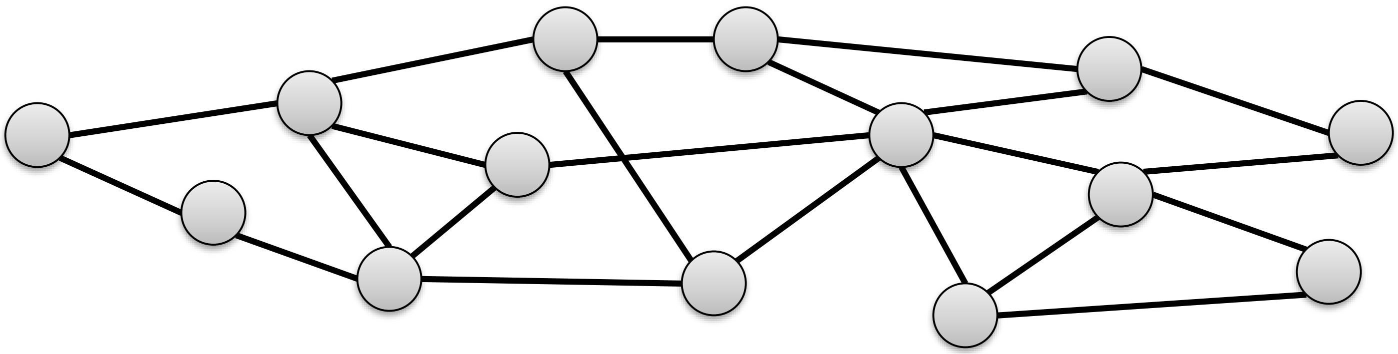
Dinur & Steurer in 2013 showed that unless $\mathrm{P} = \mathrm{NP}$, minimum set cover cannot be approximated better than by a factor $\big(1 - o(1)\big) \cdot \ln n$ in polynomial time.

- Proof is based on the so-called PCP theorem
  - PCP theorem is one of the main (relatively) recent advancements in theoretical computer science and the major tool to prove approximation hardness lower bounds
  - Shows that every language in NP has certificates of polynomial length that can be checked by a randomized algorithm by only querying a constant number of bits (for any constant error probability)

Set Cover: Special Cases

Vertex Cover: set $S \subseteq V$ of nodes of a graph $G = (V, E)$ such that

$$\forall \{u, v\} \in E, \qquad \{u, v\} \cap S \neq \emptyset.$$



Minimum Vertex Cover:

- Find a vertex cover of minimum cardinality

Minimum Weighted Vertex Cover:

- Each node has a weight
- Find a vertex cover of minimum total weight

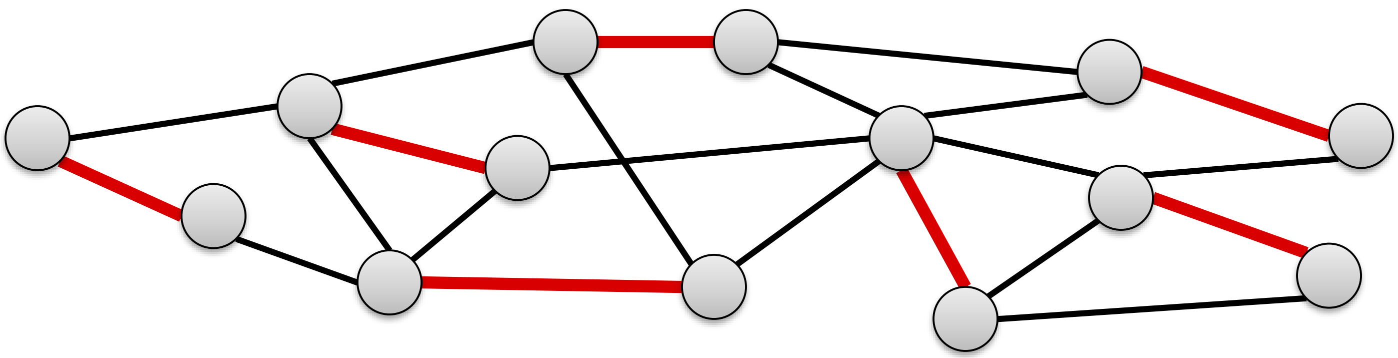
Vertex Cover vs Matching

Consider a matching M and a vertex cover S

Claim: $|M| \leq |S|$

Proof:

- At least one node of every edge $\{u, v\} \in M$ is in S
- Needs to be a different node for different edges from M



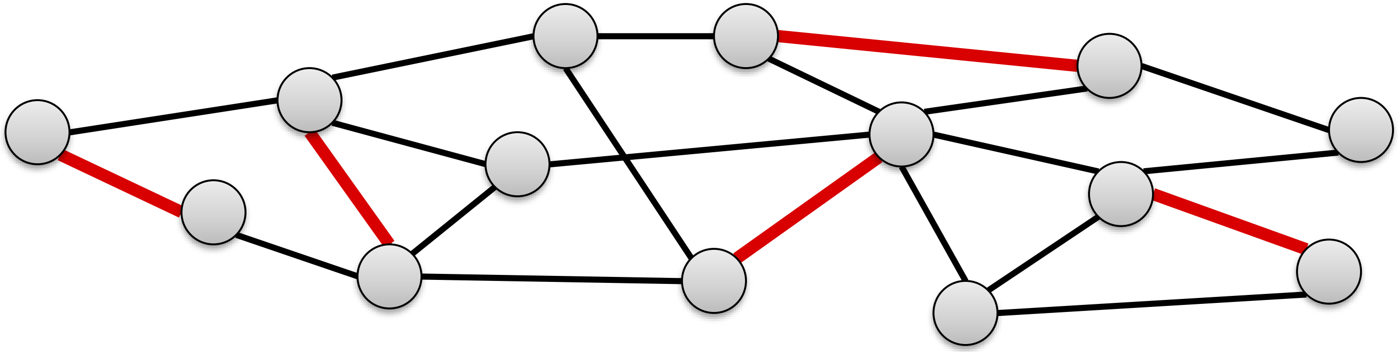
Vertex Cover vs Matching

- In the following, assume that S^* is an optimal vertex cover

Theorem: If M is a maximal matching, then $S \coloneqq \bigcup_{e \in M} e$ is a vertex cover of size $|S| \leq 2 \cdot |S^*|$.

Proof:

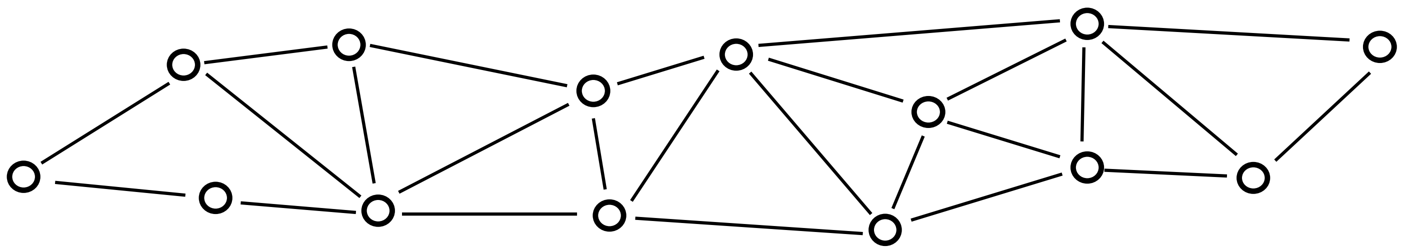
- M is maximal: for every edge $\{u, v\} \in E$, either u or v (or both) are matched



- Every edge $e \in E$ is "covered" by at least one matching edge
- Thus, the set of the nodes of all matching edges gives a vertex cover S of size $|S| = 2|M|$.

Set Cover: Special Cases

Dominating Set:
Given a graph $G = (V, E)$, a dominating set $S \subseteq V$ is a subset of the nodes V of G such that for all nodes $u \in V \setminus S$, there is a neighbor $v \in S$.



- The dominating set problem is as hard as the general set cover problem.
  - There is a simple reduction to transform every set cover instance into an equivalent dominating set instance.

Minimum Hitting Set

Given: Set of elements X and collection of subsets $\mathcal{S} \subseteq 2^X$

- Sets cover X: $\bigcup_{S \in \mathcal{S}} S = X$

Goal: Find a min. cardinality subset $H \subseteq X$ of elements such that

$$\forall S \in \mathcal{S} : S \cap H \neq \emptyset$$

Problem is equivalent to min. set cover with roles of sets and elements interchanged