Chap 35 – Approximation Algorithms

35.1 The vertex cover problem

35.2 The traveling-salesman problem

35.3 The set-covering problem

35.4 Randomization and linear programming

35.5 The subset sum problem

Approximation Algorithms

- Approximation algorithms
  - Polynomial-time
  - Suboptimal or near-optimal solutions
- Approximation ratio
  - For input size n, let C be the suboptimal solution and C^* be the optimal solution.
  - For maximization problem, $C^* \geq C \Rightarrow C^*/C \geq 1$
    For minimization problem, $C^* \leq C \Rightarrow C/C^* \geq 1$
  - A $\rho(n)$-approximation algorithm satisfies
    $\rho(n) \geq \max\{C^*/C, C/C^*\} \geq 1$
  - If it is independent of n, we use the term ρ-approximation.

35.1 The vertex cover problem

- The vertex cover problem
  - *Greedy heuristics*

    Pick up an uncovered edge, cover it with ***both*** endpoints in hope to cover more edges
  - APPROX-VERTEX-COVER(G)

    $C = \emptyset$

    $E' = G.E$

    while $E' \neq \emptyset$

        let (u, v) be an arbitrary edge of E'

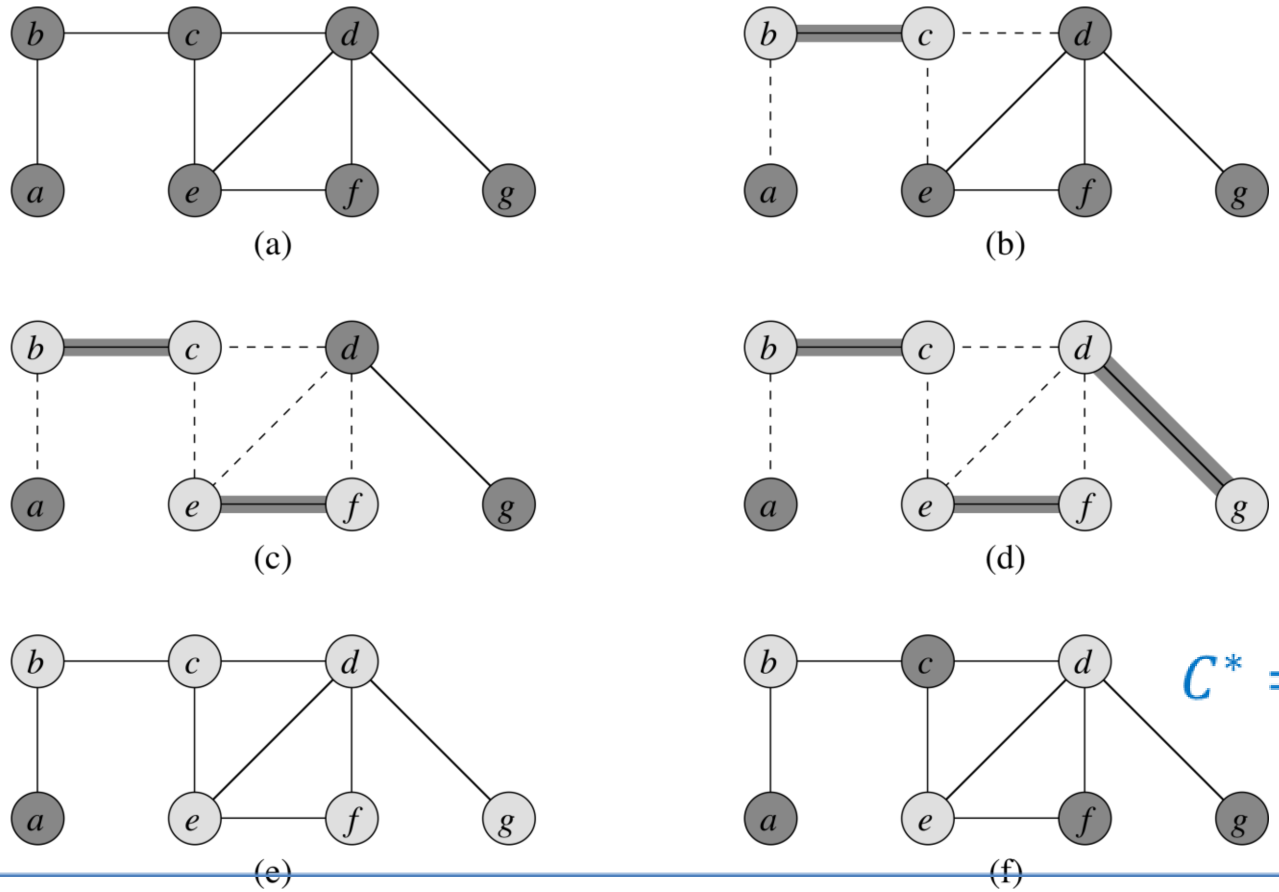
        $C = C \cup \{u, v\}$

        remove from E' every edge incident on either u or v

    return C

35.1 The vertex cover problem

- The vertex cover problem
  - Example: $C = \{b, c, d, e, f, g\}$



$C^* = \{b, d, e\}$

35.1 The vertex cover problem

- The vertex cover problem
  - **Theorem** APPROX-VERTEX-COVER is a polynomial-time 2-approximation algorithm

    Proof It runs in $O(V+E)$ time, using adjacency lists for E'

    Let A be the set of edges picked by the algorithm

    Then, no two edges in A share an endpoint.

    So, any vertex cover, including the optimal vertex cover C^*, must include at least one endpoint of each edge in A.

    Thus, $|A| \leq |C^*| \Rightarrow |C| = 2|A| \leq 2|C^*| \Rightarrow |C|/|C^*| \leq 2$

    The ratio 2 is the best possible.

    Suppose G has n isolated edges, then $C^* = n$ and $C = 2n$

35.2 The traveling-salesman problem

- TSP with triangle inequality
  - $\triangle$TSP: Assume that the cost function satisfies the triangle inequality, i.e. $c(u,w) \leq c(u,v) + c(v,w) \quad \forall u,v,w \in V$
  - **Theorem** $\triangle$TSP is NPC (Ex. 35.2-2)
  - Approx-TSP-Tour(G,c)

    Select a vertex $r \in G.V$ to be a "root" vertex

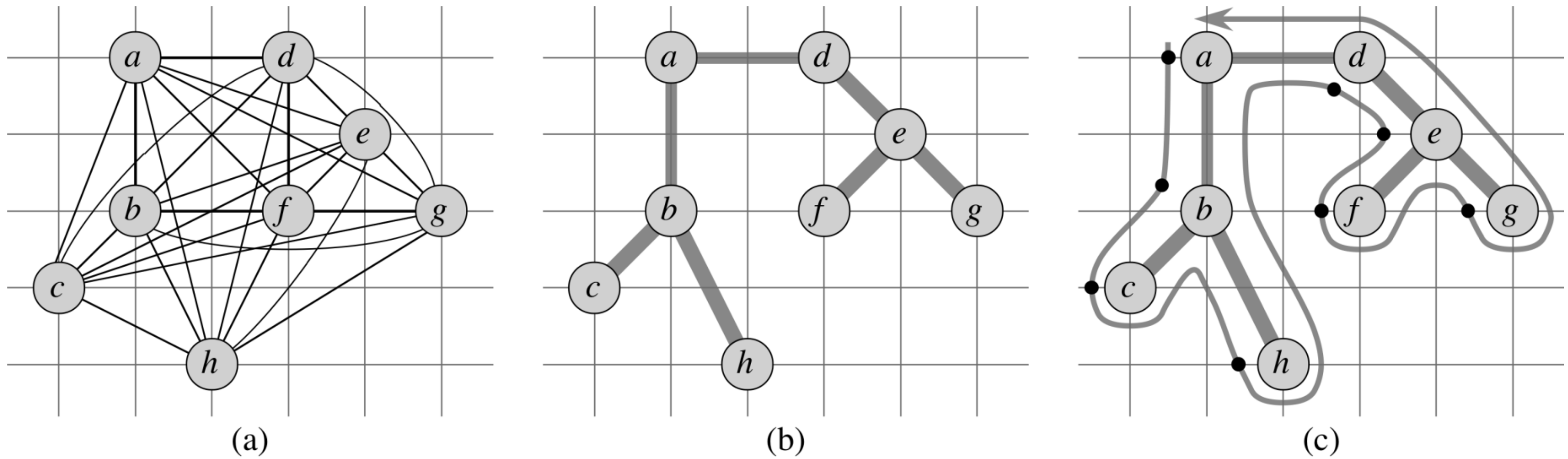
    Find a minimum spanning tree T for G from root r using MST-Prim(G,c,r)

    Let $L =$ the list of vertices visited in preorder tree walk of T

    return the Hamiltonian cycle H that visits the vertices in the order L

35.2 The traveling-salesman problem

- TSP with triangle inequality
  - Example



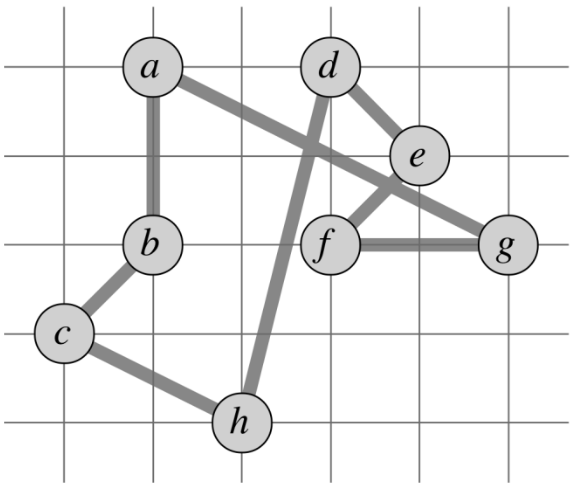
Euclidean distance MST Preorder tree walk

Prim's MST algorithm:

Starting the tree with "root" a, repeated add a light edge remained to it until the tree connects all vertices

35.2 The traveling-salesman problem

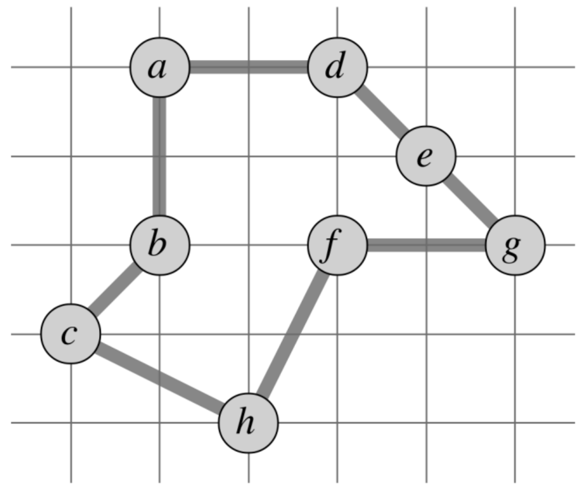
- TSP with triangle inequality
  - Example (Cont'd)



(d)

Hamiltonian circuit

Suboptimal tour



(e)

Optimal tour

35.2 The traveling-salesman problem

- TSP with triangle inequality
  - **THEOREM**

    APPROX-TSP-TOUR is a polynomial-time 2-approximation algorithm for $\triangle$TSP

    Proof It is polynomial-time: MST-PRIM runs in $O(|V|^2)$ time, and preorder tree walk runs in $O(V)$ time.

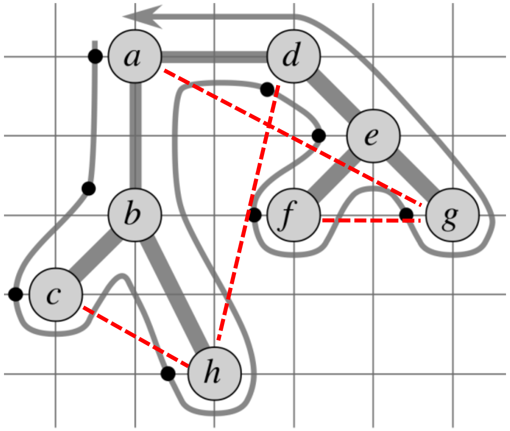
    Let H^* be an optimal tour, and T be a MST

    Since removing any edge from a tour yields a spanning tree we have $c(T) \leq c(H^* - \text{one edge}) \leq c(H^*)$.

    A ***full walk*** of T visits a vertex each time it is encountered when walking on T.

35.2 The traveling-salesman problem

- TSP with triangle inequality
  - **THEOREM** (Cont'd)



Full walk W

$a\ b\ c\ b\ h\ b\ a\ d\ e\ f\ e\ g\ e\ d\ a$

Suboptimal tour H

$a\ b\ c\ \cancel{b}\ h\ \cancel{b}\ \cancel{a}\ d\ e\ f\ \cancel{e}\ g\ \cancel{e}\ \cancel{d}\ a$

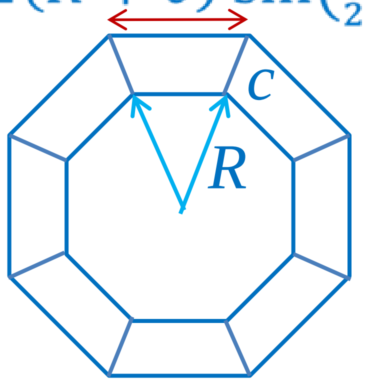
We have $c(W) = 2c(T)$, since W traverses every edge of T exactly twice. Since the suboptimal tour H is obtained from W by removing all but the 1st visit to each vertex, we have, by △ inequality, $c(H) \le c(W)$

So, $c(H) \le c(W) = 2c(T) \le 2c(H^*) \Rightarrow c(H)/c(H^*) \le 2$

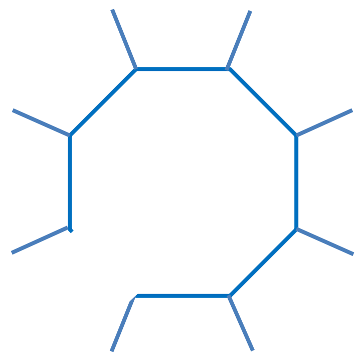
35.2 The traveling-salesman problem

- TSP with triangle inequality
  - **THEOREM** (Cont'd)

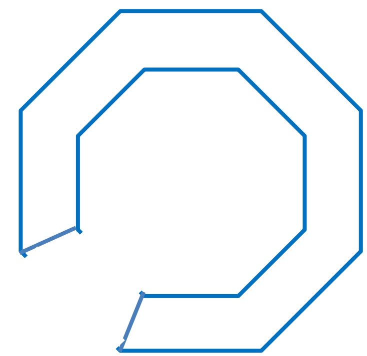
$2(R+c)\sin\left(\frac{1}{2}\cdot 2\pi/n\right)$



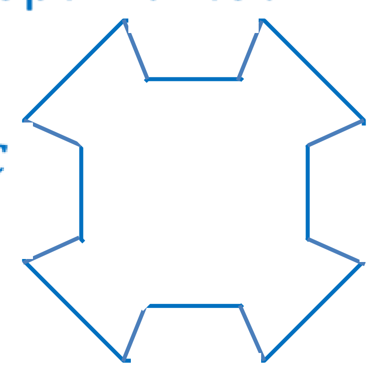
MST



suboptimal tour H



optimal tour H^*



The ratio 2 is the best possible.

$$c(H) = 2(n-1)(2R+c)\sin(\pi/n) + 2c$$

$$c(H^*) = n(2R+c)\sin(\pi/n) + nc$$

35.2 The traveling-salesman problem

- TSP with triangle inequality
  - **THEOREM** (Cont'd)

    Let $R = 1, c = 1/n^2$, then

$$\lim_{n\to\infty} c(H) = \lim_{n\to\infty}\left(2(n-1)\left(2+\frac{1}{n^2}\right)\sin(\pi/n)+\frac{2}{n^2}\right)$$
$$= \lim_{n\to\infty} 4n\sin(\pi/n) = 4\pi$$

$$\lim_{n\to\infty} c(H^*) = \lim_{n\to\infty}\left(n\left(2+\frac{1}{n^2}\right)\sin(\pi/n)+\frac{n}{n^2}\right)$$
$$= \lim_{n\to\infty} 2n\sin(\pi/n) = 2\pi$$

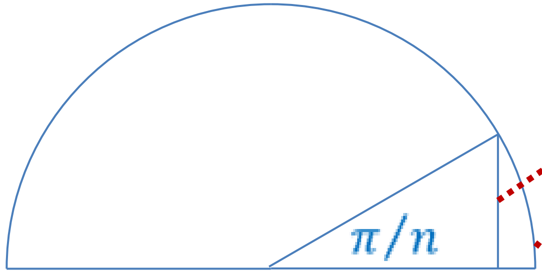
It follows that

$$\lim_{n\to\infty} c(H)/c(H^*) = 2$$

35.2 The traveling-salesman problem

- TSP with triangle inequality
  - Comment

$$\lim_{n\to\infty} n\sin(\pi/n) = \pi$$



Unit semicircle

height (sine length) $= \sin(\pi/n)$

As $n \to \infty$, sine length $=$ arc length

$\therefore$ the sum of all the n sine lengths over the unit semicircle

$=$ the perimeter of the unit semicircle

$= \pi$

35.2 The traveling-salesman problem

- General TSP
  - **THEOREM**

    If P $\neq$ NP, then for any constant $\rho \geq 1$, $\nexists$ polynomial-time ρ-approximation algorithm for the general TSP.

    Proof

    Assume to the contrary that, for some $\rho \geq 1$, APPROX-TSP is a polynomial-time ρ-approximation algorithm for TSP.

    We show that

    HAM-CYCLE $\leq_T$ TSP-APPROXIMATION

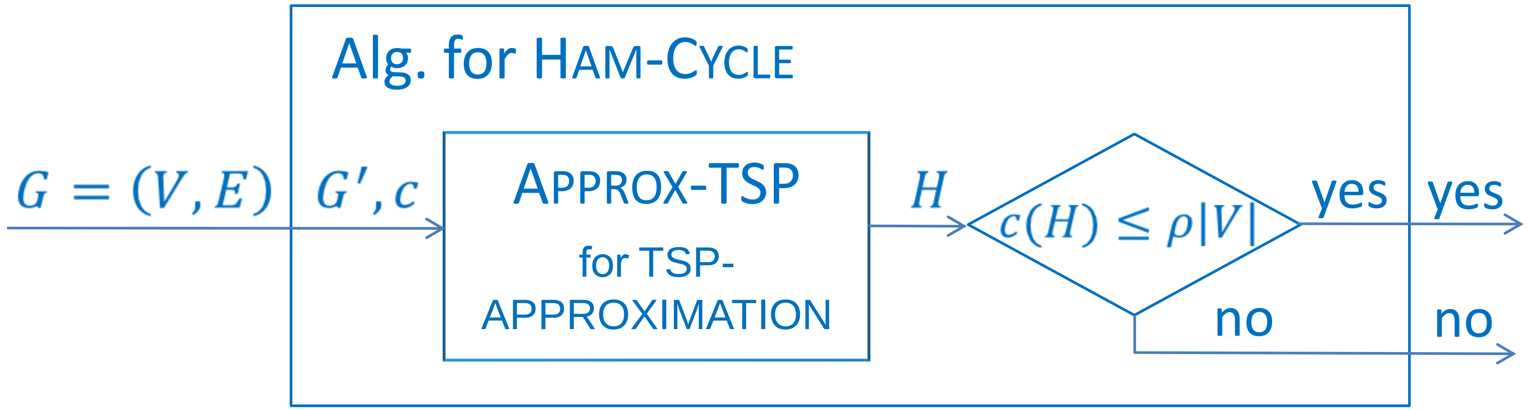
    Hence,

    HAM-CYCLE $\in$ P $\Rightarrow$ P $=$ NP. A contradiction.

35.2 The traveling-salesman problem

- General TSP
  - **Theorem** (Cont'd)

    HAM-CYCLE $\leq_T$ TSP-APPROXIMATION



Let $G = (V, E)$ be an instance of HAM-CYCLE

Reduce G to $\langle G', c\rangle$ where

$G' = (V, E')$ is a complete graph

35.2 The traveling-salesman problem

- General TSP
  - **Theorem** (Cont'd)

    $c(u,v) = 1, \text{if} (u,v) \in E$

    $\quad\quad = \rho|V| + 1, \text{otherwise}$

    Then, the reduction can be done in $O(|V|^2)$ time.

    Moreover,

    G has a Hamiltonian cycle

    $\Rightarrow$ G' has an optimal tour H^* of cost $c(H^*) = |V|$

    $\Rightarrow$ Approx-TSP returns a suboptimal tour H of cost $c(H) \leq \rho c(H^*) = \rho|V|$

    $\Rightarrow$ Ham-Cycle returns yes

35.2 The traveling-salesman problem

- General TSP
  - **Theorem** (Cont'd)

    On the other hand,

    G doesn't have a Hamiltonian cycle

    $\Rightarrow$ G' has an optimal tour H^* of cost

    $c(H^*) \geq (\rho|V| + 1) + (|V| - 1) > \rho|V|$

    $\Rightarrow$ APPROX-TSP returns a suboptimal tour H of cost

    $c(H) \geq c(H^*) > \rho|V|$

    $\Rightarrow$ HAM-CYCLE returns no

    Finally, since the reduction and the approximation can be done in polynomial time, HAM-CYCLE $\in$ P

35.3 The set-covering problem

- The set-covering problem
  - Given a finite set X and $\mathcal{F} = \{S | S \subseteq X\}$ such that
    $$X = \bigcup_{s \in F} S$$
  - Let $\mathcal{C} \subseteq \mathcal{F}$, we say that $\mathcal{C}$ covers X, if
    $$X = \bigcup_{s \in \mathcal{C}} S$$
  - Clearly, $\mathcal{F}$ covers X; so, find a minimum cover.
  - The corresponding decision problem is NPC.

    SET-COVER $= \{\langle X, \mathcal{F}, k\rangle |\ X$ has a cover of size $k\}$ // or, $\leq k$

    (See Ex.35.3-2, VERTEX-COVER $\leq_p$ SET-COVER)

35.3 The set-covering problem

- The set-covering problem
  - A greedy approximation algorithm

    Include the set that covers most elements

    GREEDY-SET-COVER$(X, \mathcal{F})$

    $U = X$

    $\mathcal{C} = \emptyset$

    while $U \neq \emptyset$ **do**

        select an $S \in \mathcal{F}$ that maximizes $|S \cap U|$

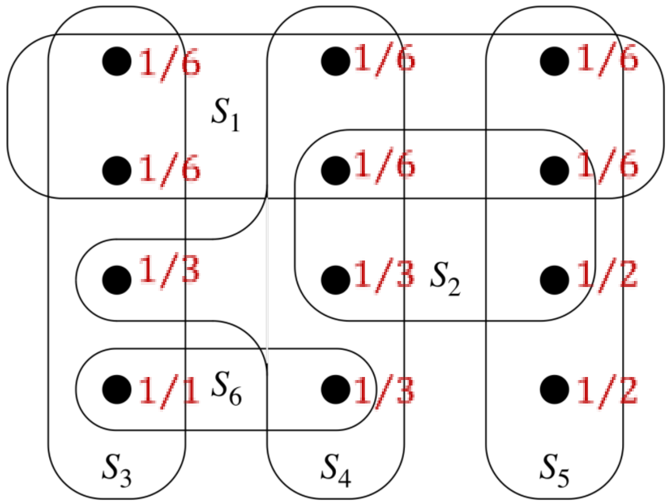
        $U = U - S$

        $\mathcal{C} = \mathcal{C} \cup \{S\}$

    return $\mathcal{C}$

35.3 The set-covering problem

- The set-covering problem
  - Example



Optimal cover $\{S_3, S_4, S_5\}$

Greedy suboptimal cover

$\{S_1, S_4, S_5, S_3\}$ or $\{S_1, S_4, S_5, S_6\}$

35.3 The set-covering problem

- The set-covering problem
  - Define $H(0) = 0$

    $H(d) = \sum_{i=1}^{d} 1/i$ be the d^{th} harmonic number, $d \geq 1$
  - **THEOREM**

    GREEDY-SET-COVER is a polynomial-time $\rho(n)$-approximation algorithm, where $\rho(n) = H(\max\{|S| : S \in \mathcal{F}\})$
  - **COROLLARY**

    GREEDY-SET-COVER is a polynomial-time $(\ln|X| + 1)$-approximation algorithm.

    Proof

    $H(\max\{|S| : S \in \mathcal{F}\}) \leq H(|X|) \leq \ln|X| + 1$ (See A.14)

35.3 The set-covering problem

- The set-covering problem
  - **Theorem** (Cont'd)

    Proof

    The loop is executed at most $\min(|X|, |\mathcal{F}|)$ times.

    The loop body can be made to run in $O(|X||\mathcal{F}|)$ time.

    So, the algorithm may run in $O(|X||\mathcal{F}| \min(|X|, |\mathcal{F}|))$ time.

    Suppose the algorithm selects $S_1, S_2, \ldots$ in that order.

    Assign a cost of 1 to each S_i and spread this cost evenly among the elements covered for the 1st time by S_i.

    For each $x \in X$, let

    $c_x =$ the cost assigned to x

35.3 The set-covering problem

- The set-covering problem
  - **THEOREM** (Cont'd)

    Thus, if $x \in X$ is covered by S_i for the 1st time, then

    $$c_x = \frac{1}{|S_i - (S_1 \cup S_2 \cup \cdots \cup S_{i-1})|}$$

    Then,

    $$|\mathcal{C}| = \#\text{ of } S \text{ selected} = \sum_{x \in X} c_x$$

    Let $\mathcal{C}^*$ be an optimal cover, we have

    $$\sum_{S \in \mathcal{C}^*} \sum_{x \in S} c_x \geq \sum_{x \in X} c_x \quad \because x \in S \text{ for at least one } S \in \mathcal{C}^*$$

35.3 The set-covering problem

- The set-covering problem
  - **Theorem** (Cont'd)

    Thus, $\displaystyle\sum_{S\in\mathcal{C}^*}\sum_{x\in S} c_x \geq |\mathcal{C}|$

    Lemma For any $S \in \mathcal{F}$, $\sum_{x\in S} c_x \leq H(|S|)$

    It follows from this lemma that

$$|\mathcal{C}| \leq \sum_{S\in\mathcal{C}^*}\sum_{x\in S} c_x \leq \sum_{S\in\mathcal{C}^*} H(|S|)$$

$$\leq |\mathcal{C}^*| \cdot H(\max\{|S| : S \in F\})$$

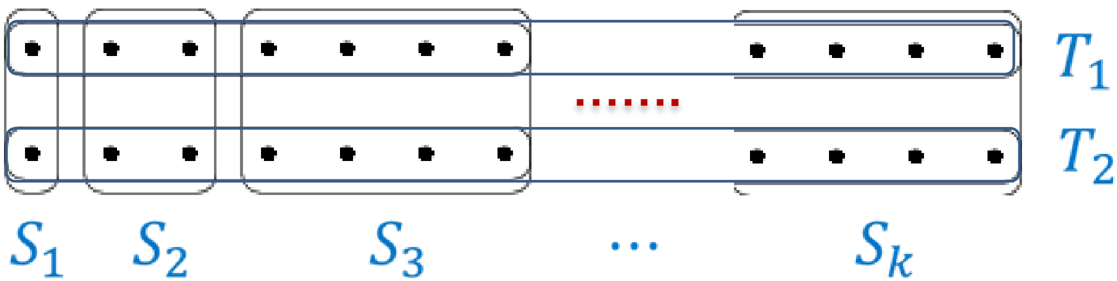
$$\Rightarrow \frac{|\mathcal{C}|}{|\mathcal{C}^*|} \leq H(\max\{|S| : S \in \mathcal{F}\})$$

35.3 The set-covering problem

- The set-covering problem
  - A standard tight example

    $X = T_1 \cup T_2 = S_1 \cup S_2 \cup \cdots \cup S_k$

    $\mathcal{F} = \{T_1, T_2, S_1, S_2, \dots, S_k\}$



$|S_i| = 2^i, \quad 1 \leq i \leq k$

$|T_1| = |T_2| = \sum_{i=1}^{k} 2^{i-1} = \sum_{i=0}^{k-1} 2^i = 2^k - 1$

$|X| = |T_1| + |T_2| = 2^{k+1} - 2$

35.3 The set-covering problem

- The set-covering problem
  - A standard tight example (Cont'd)

    Greedy suboptimal cover $\mathcal{C} = \{S_k, S_{k-1}, \dots, S_1\}$

    $\because$ First, pick S_k

    After that, $|T_1| = |T_2| = \left(2^k - 1\right) - 2^k/2 = 2^{k-1} - 1$

    So, pick S_{k-1}

    After that, $|T_1| = |T_2| = \left(2^{k-1} - 1\right) - 2^{k-1}/2 = 2^{k-2} - 1$

    So, pick S_{k-2}; and so on…

    Optimal cover $\mathcal{C}^* = \{T_1, T_2\}$

    Approximation ratio $|\mathcal{C}|/|\mathcal{C}^*| = k/2$

35.3 The set-covering problem

- The set-covering problem
  - A standard tight example (Cont'd)

    Observe that by the theorem

$$
\begin{aligned}
|\mathcal{C}|/|\mathcal{C}^*| = k/2 &\leq H(\max\{|S| : S \in \mathcal{F}\}) \\
&= H\left(2^k\right) \\
&\leq \ln 2^k + 1 \\
&= k \ln 2 + 1 \approx 0.69k + 1
\end{aligned}
$$

35.3 The set-covering problem

- The set-covering problem
  - **Lemma** For any $S \in \mathcal{F}$, $\sum_{x \in S} c_x \leq H(|S|)$

    Proof

    For $i = 1,2,\ldots,|\mathcal{C}|$

    let S_i be the i^{th} set selected by the greedy algorithm

    For any $S \in \mathcal{F}$, define

    $u_0 = |S|$

    $u_i = |S - (S_1 \cup S_2 \cup \cdots \cup S_i)|$

    $=$ # of elements in S that remain uncovered after

    selecting $S_1, S_2, \ldots, S_i$, $1 \leq i \leq |\mathcal{C}|$

35.3 The set-covering problem

- The set-covering problem
  - **Lemma** (Cont'd)

    Let k be the least index such that $u_k = 0$

    (N.B. S may or may not be S_k)

    Then,

    $$\underbrace{u_0 \geq u_1 \geq \cdots \geq u_{k-1} > u_k}_{u_{i-1} - u_i \text{ elements of } S \text{ are covered for the 1}^{\text{st}} \text{ time by } S_i} = 0 \; (= u_{k+1} = \cdots = u_{|\mathcal{C}|})$$

    Thus,

    $$\sum_{x \in S} c_x = \sum_{i=1}^{k} (u_{i-1} - u_i) \cdot \frac{1}{|S_i - (S_1 \cup S_2 \cup \cdots \cup S_{i-1})|}$$

35.3 The set-covering problem

- The set-covering problem
  - **Lemma** (Cont'd)

    Observe that

    the greedy algorithm chooses S_i rather than S

    $$\Rightarrow |S_i - (S_1 \cup S_2 \cup \cdots \cup S_{i-1})| \geq |S - (S_1 \cup S_2 \cup \cdots \cup S_{i-1})| = u_{i-1}$$

    Therefore,

    $$\sum_{x \in S} c_x \leq \sum_{i=1}^{k} (u_{i-1} - u_i) \cdot \frac{1}{u_{i-1}} = \sum_{i=1}^{k} \sum_{j=u_i+1}^{u_{i-1}} \frac{1}{u_{i-1}}$$

    $$\leq \sum_{i=1}^{k} \sum_{j=u_i+1}^{u_{i-1}} \frac{1}{j} \quad \because j \leq u_{i-1}$$

35.3 The set-covering problem

- The set-covering problem
  - **Lemma** (Cont'd)

$$\sum_{x \in S} c_x \le \sum_{i=1}^{k} \sum_{j=u_i+1}^{u_{i-1}} \frac{1}{j} = \sum_{i=1}^{k} \left(\sum_{j=1}^{u_{i-1}} \frac{1}{j} - \sum_{j=1}^{u_i} \frac{1}{j} \right)$$

$$= \sum_{i=1}^{k} \left(H(u_{i-1}) - H(u_i) \right)$$

$$= H(u_0) - H(u_k)$$

$$= H(u_0) - H(0)$$

$$= H(u_0) \qquad \because H(0) = 0$$

$$= H(|S|)$$