LECTURE 26 (November 21st)

More Approximation Algorithms

Set Cover & Randomized Rounding

Set Cover Problem Let U be a universe of n elements

Let $S_1, \dots, S_m \subseteq U$ be a family of subsets of U with associated costs $c_1, \dots, c_m$

Goal: Pick a minimum cost $\underbrace{\text{set cover}}$ of U

↳ collection of sets such that there union equals U

Note This generalizes the vertex cover problem, since

$U = \{e_1, \dots, e_m\}$ are the edges of the graph

$S_u = \{e \mid e \text{ is incident on vertex } u\}$

Set Cover is also NP-hard, but we will see an approximation algorithm for it, with $O(\log n)$ approximation, using a LP relaxation.

Integer Linear Programming Formulation

$$\min \sum_{i=1}^{m} c_i x_i$$

$$\text{s.t.} \quad \sum_{i: u \in U} x_i \geq 1 \quad \forall u \in U \qquad \text{[every element is covered]}$$

$$x_i \in \{0,1\} \quad i = 1, \dots, m \qquad x_i = \begin{cases} 1 & \text{set } i \text{ is included} \\ 0 & \text{set } i \text{ is not included} \end{cases}$$

LP relaxation

$$\min \sum_{i=1}^{m} c_i x_i$$

$$\text{s.t.} \quad \sum_{i: u \in U} x_i \geq 1 \quad \forall u \in U$$

$$0 \leq x_i \leq 1 \quad i = 1, \dots, m$$

First, let us see what approximation we can obtain by using a deterministic rounding scheme analogous to vertex cover.

Let F be the maximum frequency of any element, i.e., maximum number of subsets any element appears in.

First we solve the LP to obtain a fractional solution x^*. Note that the LP objective value

$$OPT^* = \sum_{i=1}^{m} c_i x_i^*$$

satisfies that $OPT^* \le OPT$ where OPT is the cost of optimal set cover.

Then, we round it as follows

$$x_i = \begin{cases} 1 & \text{if } x_i^* \geq 1/F \\ 0 & \text{o/w} \end{cases}$$

Then, $x = (x_1, \dots x_m)$ is an integral solution that gives a set cover.

Moreover, cost of this set cover is

$$\left[OPT \le \right] \sum_{i=1}^{m} c_i x_i \le F \sum_{i=1}^{m} c_i x_i^* = F \cdot OPT^*$$

Thus, this set cover is an F-approximation.

For vertex cover, $F = 2$, so we obtained a 2-approximation but F can be m in general, in which the approximation is trivial as the same can be achieved by including all subsets $S_1, \dots, S_m$ in the cover.

To obtain a much better approximation, we will use a randomized algorithm for rounding.

<u>Randomized Rounding for Set Cover</u>

1. Solve the LP relaxation and obtain a fractional solution x^* as before.

this will be our rounded integral solution x_i

2. For each $i = 1, \dots m$, round $x_i^* \longrightarrow 1$ with probability x_i^* (i.e. include set S_i with probability x_i^*)

3. Repeat 2 until all elements are covered.

The intuition behind this is that the higher the x_i^* value in the LP solution the higher probability of picking this set.

The above algorithm is harder to analyze so we consider a small variant :

1. Solve the LP relaxation and obtain a fractional solution x^* as before.

2. Repeat $\log n + 2$ times:
   For each $i = 1, \dots m$, round $x_i^* \longrightarrow 1$ with probability x_i^*
   (i.e. include set S_i with probability x_i^*)

   this will be our rounded integral solution x_i

3. If the final integral solution does not cover all elements or cost is more than $(4\log n + 8)$ factor of the LP solution, repeat $\boxed{2}$

To analyze this algorithm, let's see the cost of a single rounding step in $\boxed{2}$

Let $Y_i = \begin{cases} 1 & \text{if } S_i \text{ is picked} \\ 0 & \text{o/w} \end{cases}$ $\Rightarrow$ This is a random variable

After step $\boxed{2}$ finishes, $Y = (Y_1, \dots, Y_m)$ be the integral solution

Then $\mathbb{E}\left[\sum_{i=1}^{m} c_i Y_i\right] = \sum_{i=1}^{m} c_i \cdot \mathbb{E}[Y_i] = \sum_{i=1}^{m} c_i x_i^* = OPT^*$

So, the expected cost of the solution is exactly the LP objective value OPT^*

Over all the $\log n + 2$ iterations, the expected cost $\leq (\log n + 2) \cdot OPT^*$

By Markov's inequality, with probability 1/4, the cost of the final integral solution is $\leq (4\log n + 8) \cdot OPT^*$

<u>What is the probability that this integral solution is not a set cover ?</u>

Consider any fixed element of the universe, say u,

$\mathbb{P}$ [u is not covered in any one execution of the rounding step]

$$= \prod_{i: u \in S_i} \mathbb{P}[S_i \text{ is not picked}]$$

LP constraint implies $\sum_{i: u \in S_i} x_i^* \geq 1 \quad \forall u \in \mathcal{U}$

$$= \prod_{i: u \in S_i} (1 - x_i^*) \leq \prod_{i: u \in S_i} e^{-x_i^*} = e^{-\sum_{i: u \in S_i} x_i^*} \leq \frac{1}{e}$$

$$\mathbb{P}[\, u \text{ is not covered in any of the } \log n + 2 \text{ steps}] \leq \left(\frac{1}{e}\right)^{\log n + 2} \leq \frac{1}{4n}$$

By union bound

$$\mathbb{P}\left[\begin{array}{l}\exists u \text{ that is not covered in any of the} \\ \log n + 2 \text{ steps}\end{array}\right] \leq n \cdot \frac{1}{4n} = \frac{1}{4}$$

Thus, $\mathbb{P}\left[\begin{array}{l}\text{Final integral soln after step } \boxed{2} \\ \text{is a cover with cost} \leq (4\log n + 8) \cdot OPT^*\end{array}\right]$

$$\leq \frac{1}{4} + \frac{1}{4} = \frac{1}{2}$$

Thus, in expectation, step $\boxed{2}$ needs to be repeated 2 times and in the end we find a set cover whose cost is

$$\leq (4\log n + 8) \cdot \boxed{OPT^*} \longrightarrow \text{LP objective value}$$

$$\wedge|$$

$$\leq (4\log n + 8) \cdot \boxed{OPT} \longrightarrow \text{ILP objective value}$$

Thus, we obtain a $O(\log n)$ approximation.

Note: It is NP-hard to obtain a better approximation of set cover.

Hardness of Approximation

Unfortunately not all problems can be approximated beyond certain thresholds in poly-time. How do we prove that such problems are hard because these are not decision problems.

The basic idea is similar: reduce to a problem that is known to be NP-hard but one needs to take into account the approximation factors to convert it to a decision problem.

Let's see some examples.

Hardness of Traveling Salesman Problem

Traveling Salesman Problem

Given a list on n cities with distances $d(i,j)$, find the shortest tour that visits each city exactly once and returns to the initial city.

We will prove the following

Theorem For any function $f(n)$ that can be computed in polynomial time in n, there is no polynomial-time $f(n)$ approximation algorithm for the TSP on general weighted graphs unless $P=NP$.

Proof Sketch If there is an (approximating) algorithm for TSP, one can solve the Hamiltonian Cycle problem in poly- calls to the TSP algorithm.

↳ This is a decision problem:
Given a graph $G=(V,E)$, is there a Hamiltonian Cycle in graph or not.

Since Hamiltonian Cycle is a known NP-hard problem, it follows that approximating TSP is NP-hard in general.

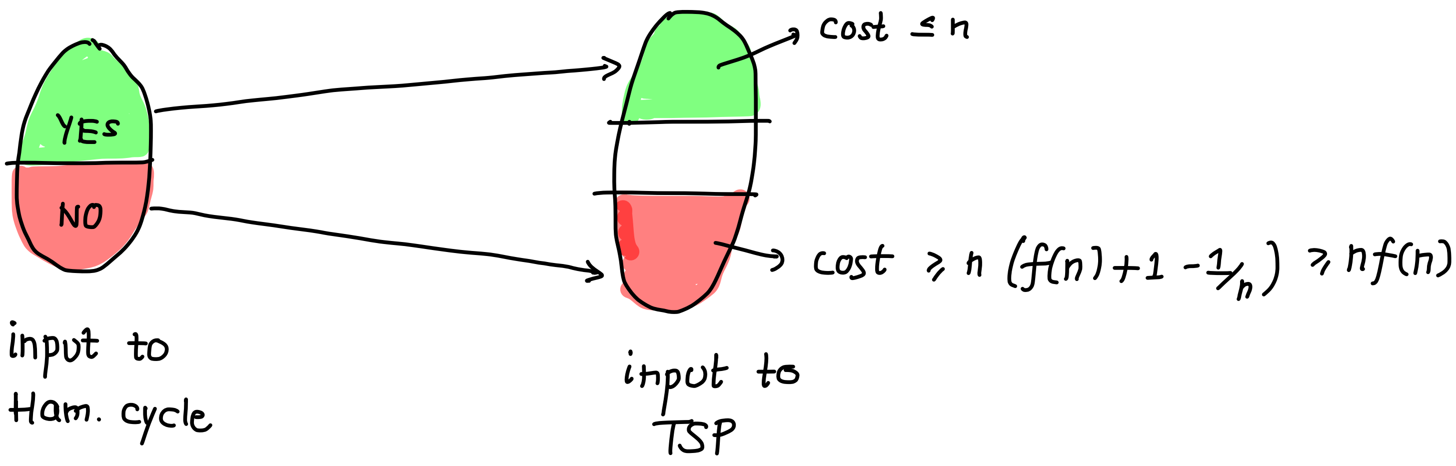
Reduction Given an instance $G=(V,E)$ for the Hamiltonian Cycle Problem we define a TSP instance as follows:

G' will be a complete graph on V &

$$d(i,j) = \begin{cases} 1 & \text{if } e \in E \\ n f(n) & \text{o/w} \end{cases}$$

(YES) If G had a Ham. cycle $\implies$ G' has a tour with cost $\leq n$

(NO) If G didn't have a Ham. cycle $\Rightarrow$ every tour in G' has cost $\geq n f(n) + n - 1$



The main property of reductions that establish hardness is the gap between the two cases.

This proves that TSP is hard to approximate with any factor $f(n)$. ∎

How to deal with problems that are even hard to approximate, such as TSP ?

Maybe our input have more structure that we are not using.

E.g. for TSP, our distances satisfy the triangle inequality in many cases of interest.

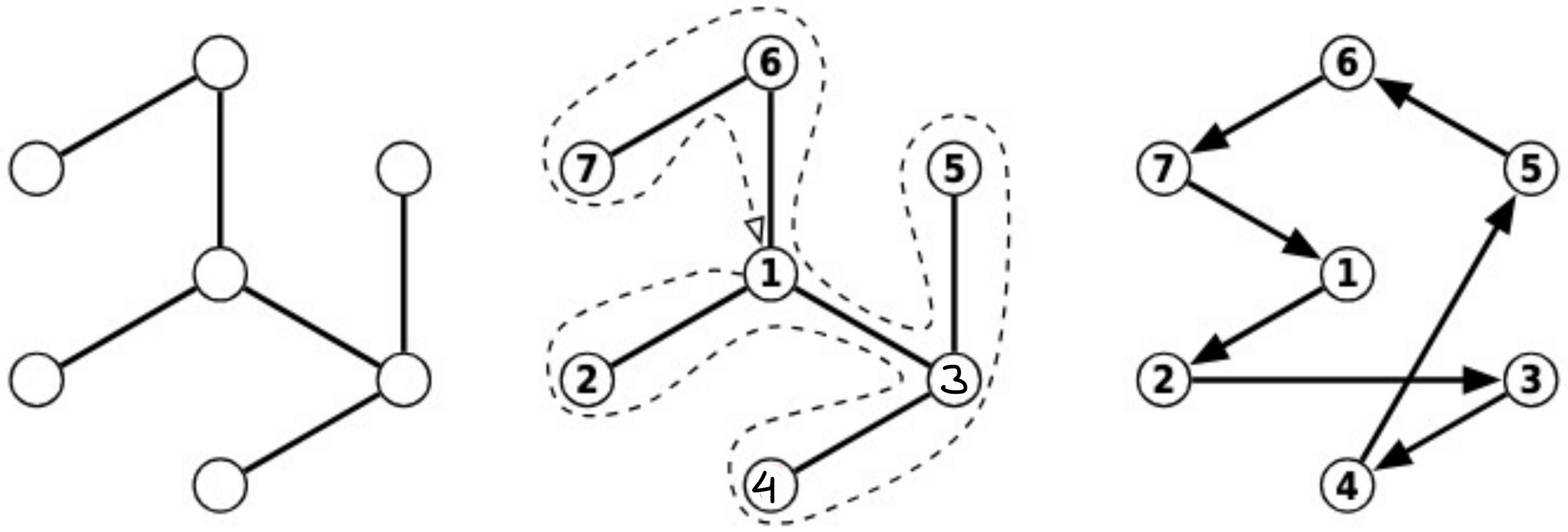
$$d(i,j) \le d(i,k) + d(k,j) \quad \forall \text{ vertices } i,j,k$$

$\underbrace{\text{In this case}}$, there is a simple 2-approximation algorithm for TSP.

This is called the Metric TSP

<u>Metric TSP algorithm</u>

1. Compute a minimum spanning tree T of the weighted input graph
2. Perform a depth-first traversal of T numbering the vertices in this order
3. Return the tour obtained by visiting the vertices according to this numbering.



$\boxed{\text{Theorem}}$ This gives a $\underbrace{\text{2-approximation}}$ to metric TSP. This can be improved with new tools.

<u>Proof</u> First, consider the "tour" computed by walking the edges of MST in the order given by depth-first search. This is not a valid TSP tour since we will visit vertices more than once. But

cost of this "tour" $\le$ 2 · cost of MST, since each edge is traversed atmost twice

The final tour is obtained by removing duplicate vertices in the "tour". This does not increase the cost because of triangle inequality, going straight only costs less.

On the other hand, cost of MST $\leq$ cost of optimal tour [Why ?]

Thus, this gives us a 2-approximation ■