Lecture 14: Overview

• Approximation Algorithms (Chapter 35)
Approximation Algorithms

- Problem:
  - Have NP-complete problem
  - Problem is important to solve
  - Can accept near optimal problem

- Approximation ratio
  - Let $C$ be the cost of solution produced by an algorithm and $C^*$ be the cost of an optimal solution
  - Define $\rho(n) \geq \max(C/C^*, C^*/C)$
  - We call such an algorithm a $\rho(n)$ approximation algorithm
Approximation Scheme

- Algorithm that takes in not only an instance of a problem but also a value of $\varepsilon > 0$ such that the scheme is a $1+\varepsilon$ approximation algorithm.

- We say that an approximation scheme is a polynomial-time approximation scheme if for any $\varepsilon > 0$ the scheme runs in time polynomial in the size $n$ of its input instance.

- Note that it can increase very quickly in $\varepsilon$.

- Fully polynomial-time approximation scheme if it is an approximation scheme and polynomial in both $1/\varepsilon$ and $n$. 
Vertex Cover Problem

- Vertex cover of a undirected graph $G=(V,E)$ is a subset $V' \subseteq V$ such that if $(u,v)$ is an edge of $G$, then either $u \in V'$ or $v \in V'$.
- Want to find one of minimal size
- NP-complete
Approx-Vertex-Cover(G)

1 C <- {}
2 E' <- E[G]
3 while E' ≠ {} 
4 do let (u,v) be an arbitrary edge of E'
5 C <- C ∪ {u, v}
6 remove from E' every edge incident on 
   either u or v
7 return C
Approx-Vertex-Cover

• Approx-Vertex-Cover is a polynomial time 2-approximation algorithm
• Let A be the edges picked in line 4
• No two edges share an endpoint
• Each edge in A has to have at least one vertex in C (|C*| ≥ |A|)
• |C| = 2|A|
• |C| ≤ 2|C*|
Traveling salesman problem

• Complete undirected graph $G=(V,E)$ that has a nonnegative integer cost $c(u, v)$ associated with each edge $(u, v)$ in $E$

• Want to find hamiltonian cycle of $G$ with minimum costs

• Let $c(A)$ denote the total cost of the edges in the subset $A \subseteq E$

• We assume the triangle inequality $c(u, w) \leq c(u, v) + c(v, w)$
Approx-TSP-Tour(G, c)

1 select vertex $r \in V[G]$ to be the root vertex
2 compute minimum spanning tree $T$ for $G$ from root $r$ using MST-Prim($G$, $c$, $r$)
3 let $L$ be the list of vertices visited in a pre-order tree walk of $T$
4 return the hamiltonian cycle $H$ that visits the vertices in the order $L$
Traveling Salesman Problem

- Approx-TSP-Tour is a polynomial-time 2-approximation algorithm for the TSP problem with the triangle inequality.
- Let H* denote an optimal tour for the given set of vertices. Since we obtain a MST tree by deleting any edge from the tour, the weight of the MST T is a lower bound on the optimal tour.
- Full walk of T lists vertices when they are first visited and when they are returned to after a visit to the subtree.
cont’d

- Since full walk traverses each edge of T exactly twice, we have
  - $c(W) = 2c(T)$
  - $c(W) \leq 2c(H^*)$
- Cost of $W$ is within a factor of two of the cost of an optimal tour
General Traveling Salesman Problem

- If \( P \neq NP \) then for any constant \( \rho \geq 1 \), there is no polynomial-time approximation algorithm with approximation ratio \( \rho \) for the general TSP.

Suppose we that for some number \( \rho \geq 1 \) there is an approx. algorithm. Assume \( \rho \) is integer (round up if necessary).

Let \( G= (V,E) \) be an instance of the hamiltonian cycle problem. Let \( G'=(V,E') \) be the complete graph on \( V \); that is

\[ E' = \{ (u,v) : u, v \in V \text{ and } u \neq v \} \]

Assign an integer cost to each edge in \( E' \) as follows

\[ c(u,v)=1 \text{ if } (u,v) \text{ in } E, \quad \rho |V| + 1 \text{ otherwise} \]
cont’d

- If the original graph has a hamiltonian cycle $H$, then the cost function $c$ assigns to each edge of $H$ has a cost of 1 so the tour cost $|V|$.
- If $G$ does not have a hamiltonian cycle, then any tour of $G'$ must use some edges not in $E$. Such a tour has a cost of at least $\rho|V|$.
Set-covering problem

- Idea:
- Have a set $X$ and a family $F$ of subsets of $X$ such that every element of $X$ belongs to at least one subset in $F$
- We say that a subset $S \in F$ covers its element
- Want to find a minimum subset of $F$ whose members cover all of $X$
Greedy-Set-Cover(X,F)

1 \( U \leftarrow X \)
2 \( C \leftarrow \{\} \)
3 while \( U \neq \{\} \)
4 \hspace{1em} do select an \( s \in F \) that maximues \( |S \cap U| \)
5 \hspace{1em} \( U \leftarrow U - S \)
6 \hspace{1em} \( C \leftarrow C \cup \{S\} \)
7 return \( C \)
cont’d

• Greedy-set-cover is a polynomial time $\rho(n)$ approximation algorithm where $\rho(n) = H(\max\{|S|:S \in F\})$ where Proof given in book. Don’t worry about it.
Projects

- Presentation dates
- Reports