CSCI 1010
Models of Computation

Lecture 21
Approximation Algorithms
Overview

• VERTEX COVER optimization problem
• 2-approximation to VERTEX COVER
• HAMILTONIAN PATH
• Traveling Salesperson optimization problem (TS) is \textbf{NP}-hard by reduction from HAM PATH
• 2-approximation to Euclidean TSP
• No \textbf{P}-time \(k\)-approximation algorithm, \(k\) fixed, exists for general TSP unless \(\textbf{P}=\textbf{NP}\).
VERTEX COVER Decision Problem

VERTEX COVER (VC)

Instance: \(<G,k>\) where \(G = (V,E)\) and \(k\) an integer.

Answer: “Yes” if there is an \(S \subseteq V, |S| = k\), such that every edge is incident on a vertex in \(S\).

Theorem VC is \(\text{NP}\)-complete.
VERTEX COVER Optimization

• Let \( G = (V, E) \) and \( VC_{\text{opt}} : \{G\} \rightarrow N, N = \{0, 1, 2, \ldots\} \) where \( VC_{\text{opt}}(G) = \) size of \textbf{minimal cover} for \( G \).

• \( VC_{\text{opt}} \) is \( \textbf{NP} \)-hard because if \( VC_{\text{opt}} \) is in \( \textbf{P} \)-time, then \( VC \) is in \( \textbf{P} \), as we show:
  
  – Compute \( VC_{\text{opt}}(G) \) on \(<G, k>\). Accept if \( VC_{\text{opt}}(G) \leq k \).
  
  – Thus, if \( VC_{\text{opt}} \) is in \( \textbf{P} \)-time, \( \textbf{P} = \textbf{NP} \). If \( \textbf{P} \neq \textbf{NP} \), then \( VC_{\text{opt}} \) is not in \( \textbf{P} \)-time.
Optimization Problems

• Minimization (maximization) problem:
  – find smallest (largest) cost solution to a problem.

• Decision problems abstract optimization problems.

• Approximation to a minimization problem is a solution close to but above the minimum.

• Roles reversed for maximization problem.
P-Time Approximation Algorithms

- For $0 < k < 1$ ($k > 1$) a P-Time $k$-approximation algorithm for an NP-hard maximization (minimization) problem is an algorithm that runs in polynomial time in the length of the input and produces a solution whose cost $C$ satisfies

$$C^* \geq C \geq kC^* \quad (C^* \leq C \leq kC^*)$$

where $C^*$ is the cost of the optimal solution.
Approximating SATISFIABILITY

SATISFIABILITY

Instances: Literals X = \{x_1, \overline{x_1}, x_2, \overline{x_2}, \ldots, x_n, \overline{x_n}\}, and clauses C = (c_1, c_2, \ldots, c_m), c_i is a subset of X.

Answer: “Yes” if x_1, x_2, \ldots, x_n have assignments such that ≥ 1 literal in each clause has value 1.

• SATISFIABILITY optimization problem is MAX SAT.

Goal: Find assignment maximizing the number of satisfied clauses.

– Can we get to ½ of max in polynomial time?
VERTEX COVER Minimization

VERTEX COVER (VC)

*Instance:* \(<G,k>\) where \(G = (V,E)\) and \(k\) an integer.

*Answer:* “Yes” if there is an \(S \subseteq V, |S| = k\), such that every edge is incident on a vertex in \(S\).

- Find a vertex cover that is closest to the smallest vertex cover. Let \(C^*\) be its size.
Approximating VERTEX COVER

Approx-VC(G) ; G = (V,E)

1. C ← φ ; C will be vertex cover*
2. E’ ← E
3. while E’ ≠ φ
4. do let (u,v) be arbitrary edge in E’
5. C ← C U {u,v} ; Add its vertices to C
6. Remove edges incident on u or v from E’
7. od
8. Return C

* Here φ is the empty set.
Approximating VERTEX COVER

- Let $C^\ast$ be size of the minimal VC for $G = (V,E)$.


- Endpoints of edges in $A$ are disjoint. Since at least $|A|$ vertices are needed to cover them, $|A| \leq |C^\ast|$.

- Thus, $|C| \leq 2|C^\ast|$, a 2-approximation. By definition, $|C| \geq |C^\ast|$. Thus, $|C^\ast| \leq |C| \leq 2|C^\ast|$.
Traveling Salesperson Problem (TSP)

- **Instance**: \((G,c), \ G = (V,E), \ c : V \times V \rightarrow N, \ N = \{0,1,\ldots\}\)
- **Objective**: Find tour (visit each vertex once) that has minimal cost \(C^* = \text{sum of costs of edges on tour.}\)

- We show TSP is **NP-hard** – reduce Hamiltonian Cycle (HC) to it. An HC visits each vertex once.

- **Hamiltonian Cycle (HC)**
- **Instance**: Graph \(G = (V,E)\)
- “Yes” **Instance**: \(G\) has a HC.
Reduce HC to TSP

• Given instance $G = (V,E)$ of HC, reduce it to an instance $G' = (V,E')$ of TSP where $E' = V \times V$ and $c(u,v) = 1$ if $(u,v)$ is in $E$ and $c(u,v) = 2$ if not.

• The instance of TSP has a tour of cost $|V|$ if $G$ has an HC and cost $> |V|$ if not.

• TSP is $\text{NP}$-hard because we can reduce HC to TSP in $\text{P}$-time.

• Thus, TSP is not $\text{P}$-time unless $\text{P} = \text{NP}$. 
2-Approximation to **Euclidean TSP**

- In **Euclidean TSP** the **triangle inequality** holds.
  \[ c(u,v) \leq c(u,w) + c(w,v) \]

**APPROX-TSP-TOUR(G,c)**

1. Select any \( r \) in \( V \) as “root”.
2. Compute minimum spanning tree \( T \) of \( G \) at \( r \).
3. Let \( L = \) list of vertices in **preorder tree walk** of \( T \)
4. **Return** Hamiltonian Cycle \( H \) that visits vertices in the order \( L \) **without repetition**.
2-Approximation to Euclidean TSP

- $L = \{a,c,g,c,h,c,a,b,d,b,e,f,e,b,a\}$
- $H = \{a,c,g,h,b,d,e,f,a\}$
2-Approximation to Euclidean TSP

• Cost $T$ of MST-tour of $(G,c)$ satisfies $T \leq 2|MST|$. Let $T^*$ be cost of optimal tour.
• Apply the triangle inequality
  $$c(u,v) \leq c(u,w) + c(w,v)$$
  to bypass repeated vertices in $L$ and form tour $H$.
• Because a spanning tree is formed by removing one edge from tour, cost of shortest tour $|MST| \leq T^*$.
• Thus $T \leq 2T^*$. By definition $T \geq T^*$. Thus, $T^* \leq T \leq 2T^*$
• Algorithm gives 2-approximation of Euclidean TSP.
General TSP

• In general TSP, triangle inequality doesn’t hold.
• **Theorem** Unless $P = NP$, for any $k \geq 1$ there is no $P$-time $k$-approximation algorithm for TSP.
• **Proof** As in HC to TSP reduction, assign cost 1 to edges in $G$ and $(k+1)|V| + 1$ to edges not in $G$.
• Thus, if $G$ has an HC, tour length = $|V|$. If not, tour length $> (k+2)|V|$.
• If a $P$-time $k$-approximate algorithm exists, it can decide HC in $P$-time. Thus, either $P = NP$ or no $k$-approximation $P$-time algorithm exists for TSP.
Overview

• VERTEX COVER optimization problem
• 2-approximation to VERTEX COVER
• HAMILTONIAN PATH
• Traveling Salesperson optimization problem (TS)
• 2-approximation to Euclidean TSP
• No P-time k-approximation algorithm, k fixed, exists for general TSP unless \( P=NP \).