CMPSCI 611: Advanced Algorithms
Lecture 20: Max Cut and Traveling Salesperson

Andrew McGregor
Approximation Ratios

Definition
The *performance ratio* of an algorithm is

$$\max_{x:|x|=n} \frac{C_{\text{alg}}(x)}{C_{\text{opt}}(x)}$$

for a minimization problem

$$\max_{x:|x|=n} \frac{C_{\text{opt}}(x)}{C_{\text{alg}}(x)}$$

for a maximization problem

where $C_{\text{alg}}(x)$ is the value of the algorithm solution on input $x$ and $C_{\text{opt}}(x)$ is the value of the optimal solution on input $x$. 
Vertex Cover

- **Input:** Graph $G = (V, E)$
- **Goal:** Find the vertex cover of smallest size?
2-approximation for Vertex Cover

Algorithm

1. \( S = \emptyset \)
2. While \( E \neq \emptyset \), pick an edge \( e = (u, v) \in E \)
   ▶ \( S \leftarrow S + u + v \)
   ▶ \( V \leftarrow V - u - v \)
3. Return \( S \)

Theorem

The above algorithm returns a 2-approximation in polynomial time.

Proof.

▶ Let \( E' \) be the set of edges chosen:

\[
\text{size of vertex cover found} = 2|E'| 
\]

▶ For any \((u, v) \in E'\), at least one of \(\{u, v\}\) is in any vertex cover:

\[
\text{size of optimal vertex cover} \geq |E'| 
\]
Outline

Set-Cover
Max Cut

- **Input:** Unweighted graph $G = (V, E)$?
- **Goal:** Find the cut $(A, B)$ that maximizes

$$|e = (u, v) \in E : u \in A, v \in B|$$

**Theorem (see notes)**

*Max-Cut is NP-Complete.*
Max Cut Approximation Algorithm

Algorithm

1. Let $A = \emptyset$, $B = V$
2. While $\exists v \in V \text{ such that switching side of } v \text{ increases size of cut:}$

   move $v$ to other side of cut

3. Return $(A, B)$

Theorem

The algorithm is a 2-approximation and runs in polynomial time.
Max-Cut Analysis

- Number of switches is at most $|E|$.
- When the algorithm terminates, let

\[
a(v) = \text{number of edges from } v \text{ that cross the cut} \\
b(v) = \text{number of edges from } v \text{ that don’t cross the cut}
\]

- Note that $a(v) \geq b(v)$ and so $\sum_v a(v) \geq \sum_v b(v)$.
- But $\sum_v a(v) + \sum_v b(v) = 2|E|$.

\[
\text{cut size} = \sum_v a(v)/2 \geq \sum_v a(v)/4 + \sum_v b(v)/4 = |E|/2
\]
Outline

Set-Cover
Metric Travelling Salesperson Problem

- **Input:** Weighted graph $G = (V, E)$ and weights $w : E \rightarrow \mathbb{R}^+$ such that for any edges $e = (u, v), e' = (v, w), e'' = (u, w)$

  \[ w_e + w_{e'} \geq w_{e''} \]

- **Goal:** Find the tour (a path that visits every node exactly once and returns to starting point) of minimum weight.

**Theorem (see notes)**

*Metric TSP is NP-Complete.*
Metric TSP Approximation Algorithm

Algorithm

1. Compute minimum spanning tree $T_{mst}$ of $G$
2. Consider a "pseudo-tour" that walks around $T_{mst}$
3. Create tour from pseudo-tour by skipping pre-visited nodes

Theorem
The algorithm is a 2-approximation.

Proof.

- Cost of pseudo-tour is twice cost of $T_{mst}$
- Cost of tour found is at most cost of pseudo-tour

\[
\text{cost(tour found)} \leq \text{cost(pseudo tour)} = 2 \cdot \text{cost}(T_{mst})
\]

- Cost of $T_{mst}$ is at most cost of optimal tour

\[
\text{cost}(T_{mst}) \leq \text{cost(optimal tour)}
\]
Outline

Set-Cover
Eulerian Tours

Definition
A Eulerian tour is a path that traverses every edge of a graph exactly once and returns back to the initial vertex.

Lemma
A graph contains an Eulerian tour iff G is connected and every vertex has even degree.
Metric TSP Approximation Algorithm

Algorithm

1. Compute minimum spanning tree $T_{mst}$ of $G$
2. Let $D$ be the nodes in $T_{mst}$ that have odd degree
3. Find minimum cost perfect matching $M$ on nodes of $D$
4. Find Euler tour of $T_{mst} + M$
5. Transform into tour by short-cutting repeated vertices.
Analysis

Theorem
The algorithm is a 3/2-approximation and runs in polynomial time.

Proof.

- Cost of tour found is at most cost of Euler tour

\[ \text{cost(tour found)} \leq \text{cost(Euler tour)} = \text{cost}(T_{mst}) + \text{cost}(M) \]

- As before, \( \text{cost}(T_{mst}) \leq \text{cost(optimal tour)} \)

- Cost of \( M \) is at most half cost of optimal tour

\[ \text{cost}(M) \leq \frac{\text{cost(optimal tour)}}{2} \]

Let \( D = \{d_1, \ldots, d_k\} \) be ordered according to optimal tour.

\[ \text{cost(optimal tour)} \geq w_{d_1,d_2} + w_{d_2,d_3} + \cdots + w_{d_{k-1},d_k} \]

\[ = (w_{d_1,d_2} + w_{d_3,d_4} + \cdots w_{d_{k-1},d_k}) + (w_{d_2,d_3} + w_{d_4,d_5} + \cdots w_{d_k,d_1}) \]
For Next Time...

- Finish reading up to Chapter 8.5.