Wrap-up
CSCI 338
Dominating Set

Dominating Set: Given a graph $G = (V, E)$ and integer $k \leq |V|$, is there a subset $V'$ of size $\leq k$, such that every vertex $\in V \setminus V'$ shares an edge with a vertex $\in V'$?
Coping with NP-Completeness

Techniques to handle NP-Complete problems:
2. Heuristics.
3. Approximation Algorithms.
Vertex Cover

Vertex Cover: Given graph, find the smallest subset of vertices such that every edge in the graph has at least one vertex in the subset. **Optimization Problem.**
**Vertex Cover – Algorithm**

**Vertex Cover:** Given graph, find the smallest subset of vertices such that every edge in the graph has at least one vertex in the subset.

**Algorithm:**

?
Vertex Cover – Algorithm

Vertex Cover: Given graph, find the smallest subset of vertices such that every edge in the graph has at least one vertex in the subset.

Algorithm:

while uncovered edge exists
    select both vertices from uncovered edge
Vertex Cover – Algorithm

Vertex Cover: Given graph, find the smallest subset of vertices such that every edge in the graph has at least one vertex in the subset.

Algorithm:

```
while uncovered edge exists
    select both vertices from uncovered edge
```

Iteration: 0
Vertex Cover – Algorithm

Vertex Cover: Given graph, find the smallest subset of vertices such that every edge in the graph has at least one vertex in the subset.

Algorithm:

while uncovered edge exists
    select both vertices from uncovered edge

Iteration: 0 1
Vertex Cover – Algorithm

Vertex Cover: Given graph, find the smallest subset of vertices such that every edge in the graph has at least one vertex in the subset.

Algorithm:

```plaintext
while uncovered edge exists
    select both vertices from uncovered edge
```

![Diagram](attachment:diagram.png)
Vertex Cover – Performance

\textbf{while} uncovered edge exists
    select both vertices from uncovered edge

Consider a set of edges, $E' \subset E$, that do not share vertices.
Vertex Cover – Performance

while uncovered edge exists
    select both vertices from uncovered edge

Consider a set of edges, $E' \subset E$, that do not share vertices. Is there a relationship between the minimum vertex cover and $|E'|$?
Vertex Cover – Performance

while uncovered edge exists
    select both vertices from uncovered edge

Consider a set of edges, $E' \subset E$, that do not share vertices. Is there a relationship between the minimum vertex cover and $|E'|$?

$$|E'| \leq \text{OPT}$$

Size of actual smallest vertex cover.
Vertex Cover – Performance

**while** uncovered edge exists  
select both vertices from uncovered edge

Consider a set of edges, \( E' \subset E \), that do not share vertices. Is there a relationship between the minimum vertex cover and \( |E'| \)?

\[ |E'| \leq \text{OPT} \]

Size of actual smallest vertex cover.

If we selected fewer than one vertex per edge, we would not have a vertex cover, because that edge would not be covered!
Vertex Cover – Performance

\textbf{while} uncovered edge exists
    select both vertices from uncovered edge

Consider a set of edges, $E' \subseteq E$, that do not share vertices. Is there a relationship between the minimum vertex cover and $|E'|$?

$|E'| \leq \text{OPT}$

Does the size of the algorithm’s output relate to a set of edges that do not share vertices?
Vertex Cover – Performance

while uncovered edge exists
    select both vertices from uncovered edge

Consider a set of edges, $E' \subset E$, that do not share vertices. Is there a relationship between the minimum vertex cover and $|E'|$?

$$|E'| \leq \text{OPT}$$

Does the size of the algorithm’s output relate to a set of edges that do not share vertices?

$$\text{ALG} = 2 \ |E'|$$
Vertex Cover – Performance

while uncovered edge exists
  select both vertices from uncovered edge

Consider a set of edges, $E' \subset E$, that do not share vertices. Is there a relationship between the minimum vertex cover and $|E'|$?

$$|E'| \leq \text{OPT}$$

Does the size of the algorithm’s output relate to a set of edges that do not share vertices?

$$\text{ALG} = 2 |E'|$$

$$\Rightarrow \text{ALG} = 2 |E'|$$
Vertex Cover – Performance

while uncovered edge exists
   select both vertices from uncovered edge

Consider a set of edges, $E' \subset E$, that do not share vertices. Is there a relationship between the minimum vertex cover and $|E'|$?

\[ |E'| \leq \text{OPT} \]

Does the size of the algorithm’s output relate to a set of edges that do not share vertices?

\[
\text{ALG} = 2 |E'| \\
\Rightarrow \text{ALG} = 2 |E'| \leq 2 \text{OPT}
\]
Vertex Cover – Performance

while uncovered edge exists
    select both vertices from uncovered edge

Consider a set of edges, $E' \subset E$, that do not share vertices. Is there a relationship between the minimum vertex cover and $|E'|$?

$$|E'| \leq \text{OPT}$$

Does the size of the algorithm’s output relate to a set of edges that do not share vertices?

$$\text{ALG} = 2 |E'|$$

$$\implies \text{ALG} = 2 |E'| \leq 2 \text{OPT} \implies \text{ALG} \leq 2 \text{OPT}$$
Vertex Cover – Performance

while uncovered edge exists
    select both vertices from uncovered edge

Consider a set of edges, $E' \subseteq E$, that do not share vertices. Is there a relationship between the minimum vertex cover and $|E'|$?

We cannot find optimal vertex covers in poly time unless $P = NP$, but this algorithm is at worst 2-times optimal.

\[ \text{ALG} = 2 \ |E'| \]

\[ \Rightarrow \text{ALG} = 2 \ |E'| \leq 2 \ \text{OPT} \Rightarrow \text{ALG} \leq 2 \ \text{OPT} \]
Vertex Cover – Performance

while uncovered edge exists
    select both vertices from uncovered edge

Consider a set of edges, $E' \subseteq E$, that do not share vertices. Is there a relationship between the minimum vertex cover and $|E'|$?

$|E'| \leq \text{OPT}$

Does the size of the algorithm’s output relate to a set of edges that do not share vertices?

$\text{ALG} = 2 |E'| \implies \text{ALG} = 2 |E'| \leq 2 \text{OPT} \implies \text{ALG} \leq 2 \text{OPT}$

Vertex Cover is approximable within the bound $2 - \frac{\log \log |V|}{2 \log |V|}$ and inapproximable within the bound 1.3606.
Independent Set

Does the approximation algorithm for Vertex Cover give an approximation algorithm for Independent Set?

Vertex Cover

Independent Set

Minimum Vertex Cover = Maximum Independent Set
Does the approximation algorithm for Vertex Cover give an approximation algorithm for Independent Set?

Minimum Vertex Cover = Maximum Independent Set

\[ \text{ALG}_{VC} \leq 2 \text{OPT}_{VC} \Rightarrow n - \text{ALG}_{VC} \geq \frac{1}{2} \text{OPT}_{IS} \]
Independent Set

Does the approximation algorithm for Vertex Cover give an approximation algorithm for Independent Set?

\[
ALG_{VC} \leq 2 \ OPT_{VC} \ \Rightarrow \ n - ALG_{VC} \geq \frac{1}{?} \ OPT_{IS}
\]
Independent Set

Does the approximation algorithm for Vertex Cover give an approximation algorithm for Independent Set?

Complete Bipartite Graph

\[ ALG_{VC} \leq 2 \cdot OPT_{VC} \implies n - ALG_{VC} \geq \frac{1}{2} \cdot OPT_{IS} \]

\[ 2n \leq 2n \quad \text{and} \quad 0 \geq \frac{1}{2} \cdot n \]
Does the approximation algorithm for Vertex Cover give an approximation algorithm for Independent Set?

**Complete Bipartite Graph**

Independent Set is inapproximable within the bound $|V|^{1-\varepsilon}$, for any $\varepsilon > 0$.

$$2n \leq 2n$$

$$0 \geq \frac{1}{?} n$$
 Complexity Hierarchy

- $P$
- $NP$
- $NP$-Complete
- $NP$-Hard

 Computability Hierarchy

- Regular
- Context-Free
- Decidable
- Turing-recognizable
- Other
TSP Approximation Algorithm

TSP: Given a weighted graph, find a least cost cycle that visits each vertex exactly once.
TSP Approximation Algorithm

TSP: Given a weighted graph, find a least cost cycle that visits each vertex exactly once.

Suppose $H$ is an $\alpha$-approximation algorithm for TSP.

I.e. $H(G) = \text{Hamiltonian Cycle } C_H$, where $\text{cost}(C_H) \leq \alpha \text{ cost}(C_{OPT})$
TSP Approximation Algorithm

TSP: Given a weighted graph, find a least cost cycle that visits each vertex exactly once.

Suppose $H$ is an $\alpha$-approximation algorithm for TSP.

I.e. $H(G) =$ Hamiltonian Cycle $C_H$, where $\text{cost}(C_H) \leq \alpha \text{cost}(C_{OPT})$

```<insert name>\rangle(G)$
   Let $C_H = H(G)$
   if $C_H == \text{null}$
     return false
   else
     return true```
TSP Approximation Algorithm

TSP: Given a weighted graph, find a least cost cycle that visits each vertex exactly once.

Suppose $H$ is an $\alpha$-approximation algorithm for TSP. I.e. $H(G) = \text{Hamiltonian Cycle } C_H$, where $\text{cost}(C_H) \leq \alpha \text{ cost}(C_{OPT})$

```python
<insert name>(G)
    Let $C_H = H(G)$
    if $C_H == \text{null}$
        return false
    else
        return true
```
TSP Approximation Algorithm

TSP: Given a weighted graph, find a least cost cycle that visits each vertex exactly once.

Suppose $H$ is an $\alpha$-approximation algorithm for TSP.
I.e. $H(G) = \text{Hamiltonian Cycle } C_H$, where $\text{cost}(C_H) \leq \alpha \text{ cost}(C_{OPT})$

```
HamiltonianCycleFinder(G)
    Let $C_H = H(G)$
    if $C_H == \text{null}$
        return false
    else
        return true
```
TSP Approximation Algorithm

TSP: Given a weighted graph, find a least cost cycle that visits each vertex exactly once.

Suppose $H$ is an $\alpha$-approximation algorithm for TSP. I.e. $H(G) =$ Hamiltonian Cycle $C_H$, where $\text{cost}(C_H) \leq \alpha \text{ cost}(C_{OPT})$

```python
HamiltonianCycleFinder(G)
    Let $C_H = H(G)$
    if $C_H == \text{null}$
        return false
    else
        return true
```

Is this a problem?
TSP Approximation Algorithm

TSP: Given a weighted graph, find a least cost cycle that visits each vertex exactly once.

Suppose $H$ is an $\alpha$-approximation algorithm for TSP.
I.e. $H(G) = \text{Hamiltonian Cycle } C_H$, where $\text{cost}(C_H) \leq \alpha \text{ cost}(C_{OPT})$

```
HamiltonianCycleFinder(G)
    Let $C_H = H(G)$
    if $C_H == \text{null}$
        return false
    else
        return true
```

Is this a problem?

Yes! Any approximation algorithm for TSP will solve the NP-Complete Hamiltonian Cycle problem!
TSP Approximation Algorithm

TSP: Given a weighted graph, find a least cost cycle that visits each vertex exactly once.

Suppose $H$ is an $\alpha$-approximation algorithm for TSP.

I.e. $H(G) = \text{Hamiltonian Cycle } C_H$, where $\text{cost}(C_H) \leq \alpha \text{ cost}(C_{OPT})$

```
HamiltonianCycleFinder(G)
    Let $C_H = H(G)$
    if $C_H == \text{null}$
        return false
    else
        return true
```

Is this a problem?

Yes! Any approximation algorithm for TSP will solve the NP-Complete Hamiltonian Cycle problem!

$\therefore \nexists \text{ poly time approx alg for TSP, unless } P = NP$
TSP Approximation Algorithm

TSP: Given a weighted graph, find a least cost cycle that visits each vertex exactly once.

Suppose $H$ is an $\alpha$-approximation algorithm for TSP.
I.e. $H(G) = \text{Hamiltonian Cycle} \ C_H$, where $\text{cost}(C_H) \leq \alpha \ \text{cost}(C_{OPT})$

Is this a problem?
Yes! Any approximation algorithm for TSP will solve the NP-Complete Hamiltonian Cycle problem!

$\therefore \ \exists \ \text{poly time approx alg for TSP, unless P = NP}$
Special Case - Metric TSP

Metric TSP: Given a complete weighted graph that satisfies the triangle inequality, find a least cost cycle that visits each vertex exactly once.

Triangle Inequality: \( \text{cost}(u, v) \leq \text{cost}(u, w) + \text{cost}(w, v) \)
Special Case - Metric TSP

Find some structure that is:

1. Easy to compute.
2. Related to TSP.
3. Lower bound on OPT.
Special Case - Metric TSP

Find some structure that is:

1. Easy to compute.
2. Related to TSP.
3. Lower bound on OPT.
Special Case - Metric TSP

Find some structure that is:
1. Easy to compute.
2. Related to TSP.
3. Lower bound on OPT.

What is this?
Special Case - Metric TSP

Find some structure that is:
1. Easy to compute.
2. Related to TSP.
3. Lower bound on OPT.

What is this?
Spanning Tree
Special Case - Metric TSP

Relationship between OPT and cost of MST?
Special Case - Metric TSP

Relationship between OPT and cost of MST?
\[ \text{cost(MST)} \leq \text{OPT} \]

How to turn MST into a cycle?
Relationship between OPT and cost of MST?
\[ \text{cost}(\text{MST}) \leq \text{OPT} \]
How to turn MST into a cycle?
What is the cost of this cycle?
Special Case - Metric TSP

Relationship between OPT and cost of MST?
\[ \text{cost(MST)} \leq \text{OPT} \]

How to turn MST into a cycle?
What is the cost of this cycle?
\[ \text{ALG} = 2 \times \text{cost(MST)} \]

Relationship between ALG and OPT?
Special Case - Metric TSP

Relationship between OPT and cost of MST?
\[ \text{cost(MST)} \leq \text{OPT} \]

How to turn MST into a cycle?
What is the cost of this cycle?
\[ \text{ALG} = 2 \times \text{cost(MST)} \]

Relationship between ALG and OPT?
\[ \text{ALG} = 2 \times \text{cost(MST)} \leq 2 \times \text{OPT} \]

Any problems?
Special Case - Metric TSP

Relationship between OPT and cost of MST?
\[ \text{cost(MST)} \leq \text{OPT} \]

How to turn MST into a cycle?
What is the cost of this cycle?
\[ \text{ALG} = 2 \times \text{cost(MST)} \]

Relationship between ALG and OPT?
\[ \text{ALG} = 2 \times \text{cost(MST)} \leq 2 \times \text{OPT} \]

How can we eliminate double visits (without messing up the cost)?
Special Case - Metric TSP

Relationship between OPT and cost of MST?
\[ \text{cost(MST)} \leq \text{OPT} \]

How to turn MST into a cycle?
What is the cost of this cycle?
\[ \text{ALG} = 2 \times \text{cost(MST)} \]

Relationship between ALG and OPT?
\[ \text{ALG} = 2 \times \text{cost(MST)} \leq 2 \times \text{OPT} \]

How can we eliminate double visits (without messing up the cost)?
Skip to next unvisited vertex. Can only decrease cost (triangle inequality).
\[ \text{dist}(u, v) \leq \text{dist}(u, w) + \text{dist}(w, v) \]
Special Case - Metric TSP

Metric TSP: Given a complete weighted graph that satisfies the triangle inequality, find a least cost cycle that visits each vertex exactly once.

\[ \text{ALG} = 2 \ \text{cost(MST)} \leq 2 \ \text{OPT} \]