Approximation Algorithms

- Finding solution to NP-complete problem is difficult.
- Two possible approaches.
  - If input is small enough, use exponential algorithm.
  - Otherwise, craft poly-time approximation algorithm.

We’ll look at approximation algorithms for
1. Vertex Cover
2. Traveling Salesman Problem
3. Set Partition Problem
Definitions

- Optimization problem on input of size \( n \).
- \( C^* \) = cost of optimal solution.
- \( C \) = cost of approximation algorithm’s solution.
- **Ratio Bound:** \( \rho(n) \) such that for input size \( n \)
  \[
  \max \left( \frac{C}{C^*}, \frac{C^*}{C} \right) \leq \rho(n) .
  \]
- **Relative Error Bound:** \( \epsilon(n) \) such that
  \[
  \frac{|C - C^*|}{C^*} \leq \epsilon(n) .
  \]
  for any \( n \).
Vertex Cover Problem

• Undirected graph $G = (V, E)$.

• Vertex cover of $G$ is $V' \subseteq V$ such that for every $(u, v) \in E$, either $u \in V'$ or $v \in V'$ (or both).

• Vertex-cover problem: find vertex cover of minimum size (optimal vertex cover).

• NP-complete (reduction from CLIQUE; see CLRS).
Example

- Find optimal vertex cover:
A possible solution

- Only solution for this graph.
- How might we approximate a solution to vertex cover problem?
**Idea**

- Choose vertices of max degree.
- Works for previous example.

![Diagram of a network with black and gray nodes connected by lines.](image-url)
Problem

• What about the following graph?

• max-degree strategy gives:
Problem

- Actual optimal solution is:

- Is there a better approximation alg.?
Approximation Algorithm

**APPROX-VERTEX-COVER**\((G)\)

1. \(C \leftarrow \emptyset\) \(\triangleright\) \(C\) to be cover
2. \(E' \leftarrow E[G]\)
3. \textbf{while} \(E' \neq \emptyset\)
4. \textbf{do} let \((u, v)\) be an arbitrary edge of \(E'\)
5. \(C \leftarrow C \cup \{u, v\}\)
6. remove from \(E'\) every edge incident on either \(u\) or \(v\)
7. \textbf{return} \(C\)
Vertex Cover Approximation Example

Proof (completed)

Q. How many $h$'s cause $x$ and $y$ to collide?

A. There are $m$ choices for each of $a_1$, $a_2$, …, $a_r$, but once these are chosen, exactly one choice for $a_0$ causes $x$ and $y$ to collide, namely

$$\sum_{i=1}^{r} a_i (x_i y_i) \pmod{m}.$$ 

Thus, the number of $h$'s that cause $x$ and $y$ to collide is $m r = |H|/m$.

October 5, 2005

Copyright © 2001-2005 by Erik D. Demaine and Charles E. Leiserson
Vertex Cover Approximation Example Cont.

Proof (completed)

Q. How many h's cause x and y to collide?

A. There are m choices for each of a_1, a_2, …, a_r, but once these are chosen, exactly one choice for a_0 causes x and y to collide, namely

\[ \frac{m^r}{m} = |H|/m. \]

October 5, 2005
Copyright © 2001-5 by Erik D. Demaine and Charles E. Leiserson

L7.15
Analysis of Vertex Cover Approximation

• Correctness
  – Only remove “covered” edges from $E'$.
  – APPROX-VERTEX-COVER returns a vertex cover.

• Running Time is $O(|V| + |E|)$. 
Further Analysis

• **Theorem**  **APPROX-VERTEX-COVER** has ratio bound 2.

• **Proof**

\[ A = \{ \text{edges picked in line 4} \} \ (A \text{ is a set}). \]
No two edges in \( A \) share an endpoint.
\[ |C'| = 2 |A|. \]
Optimal cover, \( C^* \) must include at least one end-point for each edge in \( A \).
\[ |A| \leq |C^*|. \]
Conclude that \( |C'| \leq 2 |C^*|; \)
that is, size of approximate cover is at worst twice size of optimal cover!
Traveling-Salesman Problem

- Given: **complete** undirected graph $G = (V, E)$.
- Each edge $(u, v) \in E$ has integer cost $c(u, v)$.
- Each path has an associated cost.

Traveling-Salesman Problem (TSP):

**Optimization:** find min-cost hamilt. cycle of $G$; i.e., a min-cost cycle visiting each vertex exactly once.

**Decision:** NP-complete (reduction from HAM-CYCLE, see CLRS).
Proof (completed) 
Q. How many $h$'s cause $x$ and $y$ to collide? 
A. There are $m$ choices for each of $a_1, a_2, \ldots, a_r$, but once these are chosen, exactly one choice for $a_0$ causes $x$ and $y$ to collide, namely $\mathbb{A} \cdot \mathbb{B} \cdots \mathbb{C} \cdot \mathbb{D} \cdots \mathbb{F} (x_0 y_0) \equiv a_i (x_i y_i) \pmod{m}$. Thus, the number of $h$'s that cause $x$ and $y$ to collide is $m r = |H| / m$. 

October 5, 2005 Copyright © 2001-5 by Erik D. Demaine and Charles E. Leiserson L7.15
TSP Approximation Algorithm

Suppose weights satisfy triangle inequality:

\[ c(u, w) \leq c(u, v) + c(v, w) \]

for all \( u, v, w \in V \)

\[ a + b \leq c \]

TSP is still NP-complete! However...
TSP Approximation Algorithm

\textbf{APPROX-TSP-TOUR}(G, c)

1 select a vertex \( r \in V[G] \) to be a “root” vertex
2 grow minimum spanning tree \( T \) for \( G \) from root \( r \) using \textbf{MST-PRIM}(G, c, r)
3 let \( L \) be the list of vertices visited in preorder tree walk of \( T \)
4 \textbf{return} hamiltonian cycle \( H \) that visits vertices in the order \( L \)

October 5, 2005
Copyright © 2001-5 by Erik D. Demaine and Charles E. Leiserson
Example

- Find shortest tour for:

- Find MST (with root $a$).
Example

- Pre-order walk of MST (node first, then children)

- Yielding tour:

- Total distance $\approx 24.00$ units.
Optimal Solution

- Total distance $\approx 20.44$ units.
• **Theorem:**  **APPROX-TSP-TOUR** with triangle inequality has ratio bound 2.

• **Proof:**

\[ H^* = \text{optimal tour for } G. \]
\[ T \text{ is a MST for } G \rightarrow c(T) \leq c(H^*). \]
\[ W = \text{full walk of } T. \ c(W) = 2c(T). \]
\[ c(W) \leq 2c(H^*). \]
\[ H \text{ is preorder walk of } T. \ \text{By triangle inequality, } c(H) \leq c(W) \]
\[ c(H) \leq 2c(H^*) \]

Best ratio was \( \frac{3}{2} \) for long time; now \( \epsilon \).
\( \varepsilon \)-Approximation Schemes

- Input of size \( n \) and relative error bound \( \varepsilon > 0 \).
- Returns solution with \( \frac{|C - C^*|}{C^*} \) < \( \varepsilon \).
- **Polynomial-time Approximation Scheme**
  \( O(n^{O(1)}) \) time for any constant \( \varepsilon \).
- **Fully Polynomial Time Approximation Scheme**
  Polynomial in both \( n \) and \( 1/\varepsilon \).
Partition Problem

• Given:

\[ S = \{a_1, a_2, \ldots, a_n\} \]
\[ a_1 \geq a_2 \geq \ldots \geq a_n \]

• **Problem:** Partition \( S \) into \( A \cup B \) such that \( \max (w(A), w(B)) \) is minimized.

• NP-Complete (reduction from 3D matching).

• Can we find a polynomial-time approximation scheme?
Example

\[ S: \]
\[
\begin{array}{cccccccc}
7 & 7 & 5 & 4 & 3 & 2 & 1 \\
\end{array}
\]

\[ A: \]
\[
\begin{array}{cccc}
7 & 4 & 3 & 1 \\
\end{array}
\]
\[ w(A)=15 \]

\[ B: \]
\[
\begin{array}{cccc}
7 & 5 & 2 \\
\end{array}
\]
\[ w(B)=14 \]
• Given:

\[ S = \{a_1, a_2, \ldots, a_n\} \]
\[ a_1 \geq a_2 \geq \ldots \geq a_n \]

**Approximation Scheme**

• Let \( m = \lfloor 1/\epsilon \rfloor \)

• Find optimal partition of \( S' = \{a_1, a_2, \ldots, a_m\} \)
  by exhaustive enumeration.

• Consider \( a_{m+1}, a_{m+2}, \ldots, a_n \) in turn and add to currently lighter set.
Example

$S$:  
| 7 | 7 | 5 | 4 | 3 | 2 | 1 |

- $\epsilon = 1/3$
- $m = 3$
- Partition $\{7, 7, 5\}$

$A$:  
| 7 | 5 |

$B$:  
| 7 |
Example Cont.

- Insert 4.

\[ A: \begin{align*}
7 & \quad 5 \\
\end{align*} \]

\[ B: \begin{align*}
7 & \quad 4 \\
\end{align*} \]

- Insert 3.

\[ A: \begin{align*}
7 & \quad 5 \\
\end{align*} \]

\[ B: \begin{align*}
7 & \quad 4 & \quad 3 \\
\end{align*} \]
• Insert 2.

A:

7
5
2

B:

7
4
3

• Insert 1.

A:

7
5
2
1

B:

7
4
3

• $w(A) = 15$, $w(B) = 14$.

• What is the running time for this algorithm?
Partition Problem

- **Given:**

  \[ S = \{a_1, a_2, \ldots, a_n\} \]
  
  \[ a_1 \geq a_2 \geq \ldots \geq a_n \]

- **Problem:** Partition \( S \) into \( A \cup B \) such that \( \max(w(A), w(B)) \) is minimized.

**Approximation Scheme with Relative Error < \( \varepsilon \):**

- Let \( m = \lceil 1/\varepsilon \rceil \)

- Find optimal partition of \( S' = \{a_1, a_2, \ldots, a_m\} \) by exhaustive enumeration.

- Consider \( a_{m+1}, a_{m+2}, \ldots, a_n \) in turn and add to currently lighter set.

**Time Bounds for Approximation Scheme:**

**Running Time**

- Finding optimal partition of \( S' \) takes \( O(2^m) \) time.

- Considering each of the remaining elements of \( S \) takes \( O(n) \) time.

- Total running time is

\[
O(2^m + n) = O(2^{1/\varepsilon} + n)
\]

\[ = O(n) \text{ for constant } \varepsilon \]
• Given:
  \[ S = \{a_1, a_2, \ldots, a_n\} \]
  \[ a_1 \geq a_2 \geq \ldots \geq a_n \]

• **Problem:** Partition \( S \) into \( A \cup B \) such that \( \max(w(A), w(B)) \) is minimized.

**Approximation Scheme**

• Let \( m = \lfloor 1/\epsilon \rfloor \)

• Find optimal partition of \( S' = \{a_1, a_2, \ldots, a_m\} \) by exhaustive enumeration.

• Consider \( a_{m+1}, a_{m+2}, \ldots, a_n \) in turn and add to currently lighter set.

**Proof** Approximation Scheme has Relative Error < \( \epsilon \):

**Theorem:** Partition produced by approximation scheme has relative error < \( \epsilon \).

**Proof:**

• Let \( A' \cup B' \) be an optimal partition of \( S' \).

• Assume \( w(A') \geq w(B') \).

In Case 2, Approximation is has Relative Error < \( \epsilon \) :
Proof of Approximation Scheme:

- Let $A' \cup B'$ be an optimal partition of $S'$.
- Assume $w(A') \geq w(B')$.
- Case 1
  \[ w(A') \geq \frac{1}{2} w(S) \]

Then $A = A'$, $B = B' \cup \{a_{m+1}, a_{m+2}, \ldots, a_n\}$. 
Proof of Approximation Scheme:
- Let $A' \cup B'$ be an optimal partition of $S'$.
- Assume $w(A') \geq w(B')$.
- Case 1

$$w(A') \geq \frac{1}{2}w(S)$$

Then $A = A'$, $B = B' \cup \{a_{m+1}, a_{m+2}, \ldots, a_n\}$.

In Case 1, Approximation is Exact Solution:
Claim: $A \cup B$ is optimal (Relative error = 0)
- Consider optimal solution $A^* \cup B^* = S$.
- $w(A^*) \geq w(A^* \cap \{a_1, a_2, \ldots, a_m\})$ [why?]
- $w(B^*) \geq w(B^* \cap \{a_1, a_2, \ldots, a_m\})$
- Therefore,
  $$\max (w(A^*), w(B^*)) \geq \max (w(A'), w(B'))$$
  $$= w(A')$$
  $$= w(A).$$
- Hence, $A \cup B$ is optimal.
Proof of Approximation Scheme:

- Let \( A' \cup B' \) be an optimal partition of \( S' \).
- Assume \( w(A') \geq w(B') \).

In Case 2, Approximation is has Relative Error \(< \varepsilon>\):

- Case 2

\[
    w(A') \leq \frac{1}{2}w(S)
\]

\[
    |w(A) - w(B)| \leq a_{m+1}
\]

\[
    w(A) + w(B) = w(S)
\]

\[
    2 \max (w(A), w(B)) \leq w(A) + w(B) + a_{m+1}
\]

\[
    \leq w(S) + a_{m+1}
\]

\[
    \Rightarrow \quad C = \max (w(A), w(B)) \leq \frac{w(S) + a_{m+1}}{2}
\]
In Case 2, Approximation has Relative Error < \( \varepsilon \):

**Case 2** \( \mathcal{A}' \), \( w(\mathcal{A}') \leq \frac{1}{2} w(\mathcal{S}) \)

\[
\text{Problem: } \text{Partition } \mathcal{S} \text{ into } \mathcal{A} \cup \mathcal{B} \text{ such that } C = \max \left( w(\mathcal{A}), w(\mathcal{B}) \right) \leq \frac{w(\mathcal{S}) + a_{m+1}}{2} \text{ minimized.}
\]

Relative Error \[
= \frac{C - C^*}{C^*} = \frac{w(\mathcal{S}) + a_{m+1}}{2} - \frac{w(\mathcal{S})}{2}
\]

\[
= \frac{a_{m+1}}{w(\mathcal{S})} \leq \frac{a_{m+1}}{(m + 1)a_{m+1}} = \frac{1}{m + 1} \leq \varepsilon \] 

Since \( m = \left\lfloor 1/\varepsilon \right\rfloor \)

\[
\text{approx. alg. for partition has relative error } < \varepsilon!
\]