Euclidean TSP

Point set \( P \)

\( \omega(pq) = |pq| \)

Property 1 (Triangular Inequality):

\[ |ac| \leq |ab| + |bc| \]

Property 2 (Extension of Property 1):

\[ |ab| \leq |e_1| + |e_2| + \ldots + |e_m| \]

First Alg.

- Find \( \text{MST}(P) \)
- Duplicate edges of \( \text{MST}(P) \)
- Find an Eulerian tour \( E \)
- Convert \( E \) to a tour \( T \) by going through vertices of \( E \) while skipping the vertices that already visited.

Eulerian tour: a closed walk which visits each vertex of the graph exactly once.

Theorem: a graph has an Eulerian tour iff all vertices are of even degree.
Theorem: The above alg. is a 2-approximation alg. for ETSP.

Proof:

Let OPT be an optimal tour. By removing any edge from OPT we obtain a spanning path $T'$ which is a spanning tree as well. Thus:

\[ T' \leq \text{OPT} \implies \text{MST} \leq \text{OPT} \quad (1) \]

Abuse the notation for weight:

\[ \text{MST} \leq T' \]

The weight of $E$ is two times the weight of MST because $E$ contains two copies of each edge in MST.

\[ E = 2 \text{ MST} \]

Since we obtain $T$ by shortcutting the edges of $E$, the cost of $T$ does not exceed the cost of $E$ (remember the triangle inequality):

\[ \implies T \leq E = 2 \text{ MST} \quad (2) \]

Finally \( (1)(2) \):

\[ T \leq E = 2 \text{ MST} \leq 2 \cdot \text{OPT} \]

\[ \implies T \leq 2 \cdot \text{OPT} \]
Second Alg.
- Find MST($P$)
- Let $V'$ be the vertices of odd degree in MST($P$)
- Compute a minimum perfect matching $M$ for $V'$
- Add edges of $M$ to MST($P$)
- Find an Eulerian tour $E$ in $M +$ MST($P$)
- Convert $E$ to a tour $T$

Theorem: The above alg. is a $\frac{3}{2}$-approximation for the ETSP.

Let OPT be the an optimal tour and let $T'$ be the tour obtained from OPT by shortcutting the vertices of $V \setminus V'$.

$T' \leq \text{OPT}$

$T'$ contains two perfect matching for $V$, say $M_1$ and $M_2$

$M$ is a minimum perfect matching and smaller than both $M_1$ and $M_2$

$M \leq M_1$ and $M \leq M_2 \Rightarrow 2M \leq M_1 + M_2 = T'$

$\Rightarrow M \leq \frac{T'}{2} \leq \frac{\text{OPT}}{2}$
\[ E = M + HST \leq \frac{OPT}{2} + OPT = \frac{3}{2} OPT \]
Load Balancing:

Given \( n \) jobs \( J_1, \ldots, J_n \) and \( m \) machines \( M_1, \ldots, M_m \).

Each job \( J_i \) needs time \( t_i \) to be done.

Assign jobs to machines such that the total time until all jobs are finished is minimized.

**First Alg. (Greedy)**

\[
\text{for } i \leftarrow 1 \text{ to } n \\
\text{ assign } J_i \text{ to } M_k \text{ of minimum load}
\]

**Theorem:** The above alg. is a 2-approximation alg. for load balancing prob.

Let \( T \) be the time required by alg. and \( \text{OPT} \) be the optimal time.

\[
\text{OPT} \geq t_{\text{max}} \quad \text{where } t_{\text{max}} = \max(t_i)
\]

\[
\text{OPT} \geq \frac{1}{m} \sum_{i=1}^{n} t_i
\]

At the end of alg., let \( M_{\text{f}} \) be the machine which maximizes the load.

Let \( J_{\ell} \) be the last job assigned to \( M_{\text{f}} \).

\[
T = T_f + t_{\ell} \leq \frac{1}{m} \sum_{i=1}^{n} t_i + t_{\text{max}} \leq \text{OPT} + \text{OPT}
\]

\[
\Rightarrow T \leq 2 \cdot \text{OPT}
\]
How to improve:

\[
\begin{array}{c}
\frac{1}{v_1} & \frac{1}{v_2} & \frac{1}{d_3} & \frac{1}{d_4} & \frac{1}{d_5} \\
M_1 & M_2 & M_3 & M_4
\end{array}
\]

**Assignment 1**

\[
\begin{array}{c}
M_1 & \square & 1 & \cancel{2} \\
M_2 & \square & \cancel{1} \\
M_3 & \square & \cancel{1} \\
M_4 & \square & \cancel{1} \\
\end{array}
\]

\(T = 3\)

**Assignment 2**

\[
\begin{array}{c}
M_1 & \square & \cancel{2} \\
M_2 & \square & \cancel{1} \\
M_3 & \square & \cancel{1} \\
M_4 & \square & \cancel{1} \\
\end{array}
\]

\(T = 2\)

Observation: It's better to assign larger jobs first.

**Second Alg. 1**

\[
L \leftarrow \text{decreasing ordered list of jobs}
\]

For \(i = 1 \text{ to } n\)

assign \(L(i)\) to \(M_k\) of minimum load

**Theorem**: The above alg. is a \(\frac{3}{2}\)-approximation alg. for load balancing problem.

Assume \(t_1 \geq t_2 \geq \ldots \geq t_n\)

- if \(n \leq m\) then the alg. is optimal
- if \(n > m\) then \(\text{opt} \geq t_m + t_{m+1}\)

Let \(M_f\) be the machine with maximum load and assume \(J_e\) is the last job assigned to \(M_f\).

\[
\begin{array}{c}
t_1 \\
\ldots \\
t_m \\
t_{m+1} \\
t_n
\end{array}
\]

\[
t_c \leq t_m \\
t_c \leq t_{m+1} \quad \Rightarrow \quad 2t_c \leq t_m + t_{m+1} \quad \Rightarrow \quad t_c \leq \frac{t_m + t_{m+1}}{2}
\]

\[
M_f
\]

\[
T = T_f + t_c \leq \frac{1}{m} \sum_{i=1}^{m} t_i + \frac{t_m + t_{m+1}}{2} \leq \text{opt} + \frac{\text{opt}}{2}
\]

\(\Rightarrow\) \(T \leq \frac{3}{2} \cdot \text{opt}\)
**Vertex-Cover Problem**

$G = (V, E)$, a vertex-cover is a subset $C \subseteq V$ s.t. for each edge $(a, b) \in E$, either $a \in C$ or $b \in C$ (or both).

$C$ covers all the edges of $G$!!

**Vertex-Cover Problem**: compute a cover $C^*$ of minimum size.

**First Alg.**

\[
\begin{align*}
C & \leftarrow \emptyset \\
\text{while } E \neq \emptyset & \text{ do} \\
(a, b) & \leftarrow \text{arbitrary edge in } E \\
C & \leftarrow C \cup \{a, b\} \\
E & \leftarrow E \setminus \{\text{all edges incident on } a \text{ or } b\}
\end{align*}
\]

return $C$

Running Time = $O(V + E)$

**Theorem**: The above alg. is a 2-approx alg. for **vertex-cover problem**.

Let $M$ be the set of edges selected in the while loop.

Let $C^*$ be an optimal vertex cover.

for each edge $(a, b) \in M$, $C^*$ contains either $a$ or $b$. $\Rightarrow |C^*| \geq |M|$

We add both of $a$ and $b$ to $C$ $\Rightarrow |C| \leq 2|A|$

$\Rightarrow C \leq 2|A| \leq 2|C^*|$

$\Rightarrow$ Any maximal matching in $G$ is a 2-approx for **vertex-cover problem**.
Second Alg.

\[ C \leftarrow \emptyset \]

while \( E \neq \emptyset \) do

\[ a \leftarrow \text{vertex with maximum degree in current graph} \]

\[ C \leftarrow C \cup \{a\} \]

\( E \leftarrow E \setminus \text{all edges incident on } a \)

return \( C \).

What is the approximation ratio \( \alpha \)?

\[ \alpha = \Omega \left( \log n \right) \]

\[ \begin{align*}
|B| &= |R| + \frac{|R|}{2} + \frac{|R|}{3} + \cdots + \frac{|R|}{i} \\
&= |R| \left( \sum_{i=1}^{\infty} \frac{i}{i} \right) = |R| \log |R| \\
C &= B \\
C^* &= R \\
\Rightarrow \frac{|C|}{|R|} &= \log |R| = \Theta(\log n) \quad \Rightarrow \alpha \geq \log n
\end{align*} \]
**K-Clustering:**

For a point set \( P \), diameter of \( P \), \( d(P) \), is defined as the maximum distance between any pair of points in \( P \).

\[
d(P) = \max \{|pq| : p, q \in P\}
\]

Given a point set \( P \), we want to partition \( P \) into \( k \) clusters such that the diameter of clusters is minimized.

\[ \Rightarrow \text{we want to minimize the maximum diameter.} \]

Example:

\[ k = 3 \]

![Diagram of K-clustering example]

**Voronoi Diagram (VD):**

Given a point set \( S \), \( VD(S) \) is defined as depicted in the picture.

**Algorithm**

Idea: pick \( k \) points of \( P \) as \( S \)

- Compute \( VD(S) \)
- Assign the remaining points to each Voronoi cell.
Algorithm

\[ S \leftarrow \text{arbitrated point of } P \]
for \( i = 2 \) to \( k \)
\[ x \leftarrow \text{point of } P \text{ which is furthest from } S \]
\[ S \leftarrow S \cup \{ x \} \]
Assign the points in \( P \setminus S \) to their closest point in \( S \).

Theorem: The above alg. is a 2-approx alg. for the \( k \)-clustering problem.

Let \( y \) be the next element that we have chosen by repeating the for loop \((k+1)^{th} \) element\) and let \( r \) be the distance of \( y \) to its closest point in \( S \).

All points in cluster \( s \) are at distance at most \( r \) of \( s \), then the diameter of \( s \) is at most \( 2r \).

We have \( k+1 \) points such all of them are at distance at least \( r \) from each other (in each iteration the furthest point distance is decreasing) and by region hole principle two of them fall in the same cluster in OPT. So OPT has diameter at least \( r \).