Alon Mizrahi 205978042
Roy Ben Dov 311120380

Metric TSP Approximations

We discuss the Metric TSP case which is a special case of TSP where the Triangle Inequality holds ,the distances are symmetric , and you can reach from each node to another node. That is when $\forall a,b,c\ :\ d(a,c) \le d(a,b) + d(b,c)$ and $\forall a,b\ :\ d(a,b) = d(b,a)$.

The TSP problem asks the following question:
"Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city and returns to the origin city?"

The problem is equivalent to finding a minimum-weight Hamiltonian cycle in a complete, weighted graph, which is NP hard.
In this article we will tackle this problem, therefore for every algorithm the input will always be a complete, weighted graph thats holds the Triangle Inequality: $G = (V,E)$ with weights $w_{i,j}\ \forall e(i,j) \in E \ \wedge\ e(i,j) = e(j,i)$.

Also denote for every given graph $G = (V,E)$, $d(E) := \sum_{(u,v)\in E} w_{u,v}$.

This will also hold for graphs, i.e. $d(G) = d(E)$.

We will present two approximations:

- A well known 2 approximation.
- A $\frac{3}{2}$ approximation, called Christofides' Algorithm, published by Nicos Christofides in 1976.

We will display and prove each algorithm, as well as analyze time complexities.
We will also propose a Java implementation for the 2 approximation along with practical testing and analyzed results.

2 Approximation

1. Construct a minimum spanning tree MST out of G.
2. Perform a DFS tour in MST from any starting vertex, hitting every edge exactly twice. Let T be the tour given by the DFS.
3. Return the tour T^*, which is T with taking “shortcuts”: we remove every vertex from T which does not appear in its first time.

Claim:
Let OPT be the Hamiltonian cycle with the minimum weight in G. Then $d(T^*) \leq 2 \cdot d(OPT)$. From now on we will denote $OPT^* = d(OPT)$

Proof:
Claim 1: $d(MST) \leq OPT^*$.
Assume by contradiction that $OPT^* < d(MST)$. OPT, by definition, is a cycle in G. then $OPT \setminus \{e\}$ for some $e \in OPT$ is a tree which applies $d(OPT \setminus \{e\}) < OPT^*$. We now get $d(OPT \setminus \{e\}) < d(MST)$, therefore MST is not a minimum spanning tree, a contradiction.

Claim 2: $d(T) = 2 \cdot d(MST)$.
This holds due to the way our specific DFS works (recursively adding non-visited vertices and <u>returning</u> to the vertices after the recursion ends). We will visit each edge in MST <u>twice</u>.

Claim 3: $d(T^*) \leq d(T)$.
This holds due to the Triangle Inequality property of G. Going from u to v is cheaper than going from u to t to v.

Finally, we get $d(T^*) \leq d(T) = 2 \cdot d(MST) \leq 2 \cdot OPT^*$. □

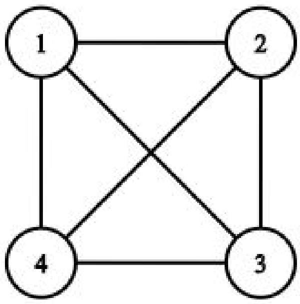
Time complexity:

1. Construct MST - Kruskal’s algorithm: $O(|E| \cdot \log|V|) = O(n^2 \log(n))$.
2. DFS - $O(|E| + |V|) = O(n)$.
3. Construct T^* with shortcuts in T - $O(n)$.

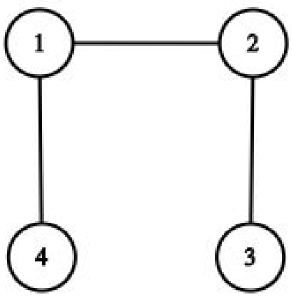
Total: $O(n^2 \log(n))$.

Simulation

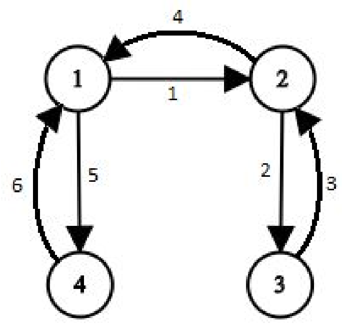
Consider the complete graph:



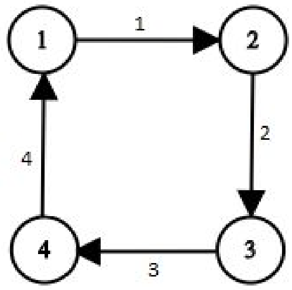
Step 1: Construct MST.



Step 2: Run a DFS tour.



Step 3: Take shortcuts.



1.5 Approximation

1. Construct a minimum spanning tree MST out of G.
2. Find a minimum weight perfect matching P in G' between the vertices that have odd degree in MST. (G' is the subgraph of G where its set of vertices is only the vertices with odd degree in MST).
3. Find an Eulerian cycle T in $MST \cup P$ (note that $MST \cup P$ in this context is a *disjoint union*, therefore it might result in a multigraph).
4. Return the tour T^*, which is T with taking “shortcuts” (Same as for the 2 approximation).

Claim: Let OPT be the minimum weight Hamiltonian cycle in G. Then $d(T^*) \leq 1.5 \cdot OPT^*$.

Proof:
Claim 1: $d(MST) \leq OPT^*$.
Proved in the previous section.

Claim 2: The number of vertices with odd degrees in MST is even.
Proof:

By the Handshaking Lemma, we know that $\sum_{v \in V} deg(v) = 2 \cdot |E|$.

Also, $\sum_{v \in V} deg(v) = \sum_{v \in V \text{ s.t. } deg(v) \text{ is even}} deg(v) + \sum_{v \in V \text{ s.t. } deg(v) \text{ is odd}} deg(v)$.

We know that $\sum_{v \in V \text{ s.t. } deg(v) \text{ is even}} deg(v)$ is even, because we sum even numbers.

Therefore, $\sum_{v \in V \text{ s.t. } deg(v) \text{ is odd}} deg(v) = 2 \cdot |E| - \sum_{v \in V \text{ s.t. } deg(v) \text{ is even}} deg(v)$, is a difference between two even numbers, which is even.

- Note that by the previous theorem, there must exist a perfect matching in G', because it’s a subgraph of a complete graph with an even number of vertices.

Claim 3: $d(P) \leq \frac{1}{2} \cdot OPT^*$.
Proof:
Let O be the set of vertices with odd degree in MST .
Let N^* be the minimum-weight Hamiltonian cycle in $G' = (O,\ E_O)$.
We can decompose N^* into two matchings $N_1,\ N_2$ by taking the edges of N^* alternately.
Therefore: $d(P) \leq min\{d(N_1), d(N_2)\}$.
The inequality is true because P is a minimum-weight perfect matching.
Also, $d(N_1) + d(N_2) = d(N^*)\ \rightarrow\ min\{d(N_1), d(N_2)\} \leq \frac{1}{2}d(N^*)$ therefore $d(P) \leq \frac{1}{2}d(N^*)$.
Lastly, we're in the case where the triangle inequality holds, therefore $d(N^*) \leq OPT^*$. This is true because N^* is an the optimal solution for a subgraph of G, and adding vertices will only add more weight to the solution.

- Note that there now exists an Eulerian cycle in $MST \cup P$ because all the vertices in that graph have even degrees.

Claim 4: $d(T) = d(MST) + d(P)$.
This is true by the definition of Eulerian cycles. In an Eulerian cycle, we cross every edge exactly once.

We now get $d(T) = d(MST) + d(P) \leq OPT^* + \frac{1}{2}OPT^* = 1.5 \cdot OPT^*$.

At last, we previously proved that $d(T^*) \leq d(T)$ because of the triangle inequality we assume. Finally we get $d(T^*) \leq 1.5 \cdot OPT^*$. □

Time complexity:

1. Construct MST - Kruskal's algorithm: $O(|E| \cdot \log |V|) = O(n^2 \log(n))$.
2. Find minimum perfect matching - $O(|E| \cdot |V|^2) = O(n^4)$.
3. Eulerian cycle - $O(|E_{MST \cup P}|) = O(n)$.
4. Construct T^* with shortcuts in T - $O(n)$.

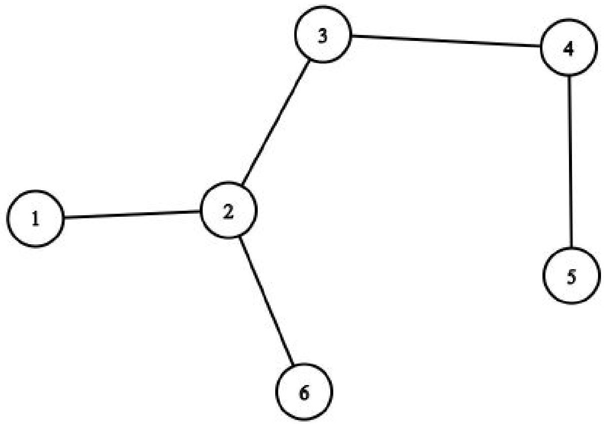
Total: $O(n^4)$.

- The minimum perfect matching can be achieved by using an adaption of Blossom algorithm, which is out of the scope of this article. It operates in $O(|E| \cdot |V|^2)$.

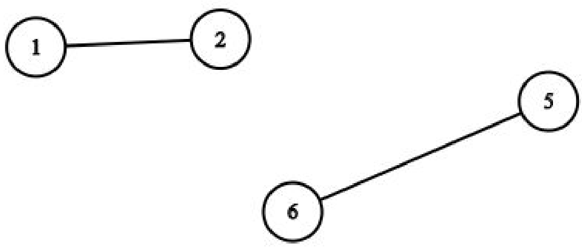
Simulation

Consider a complete graph of 6 vertices, and the next MST .

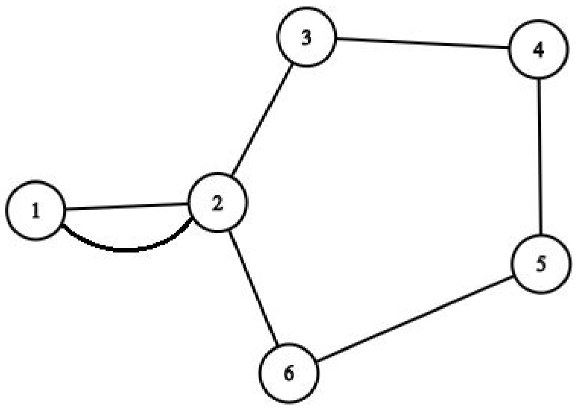
Step 1: Construct MST.



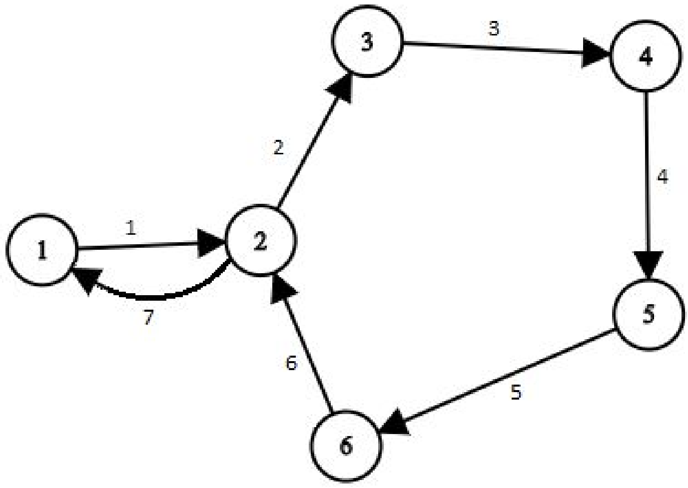
Step 2: Find minimum weight matching P in G' with odd vertices of MST .



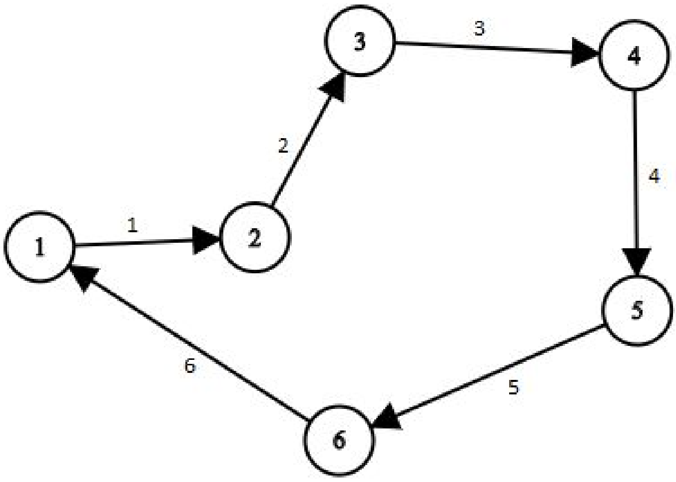
The next graph is $MST \cup P$ (in this case we get a multigraph).



Step 3: Find Eulerian cycle in $MST \cup P$.



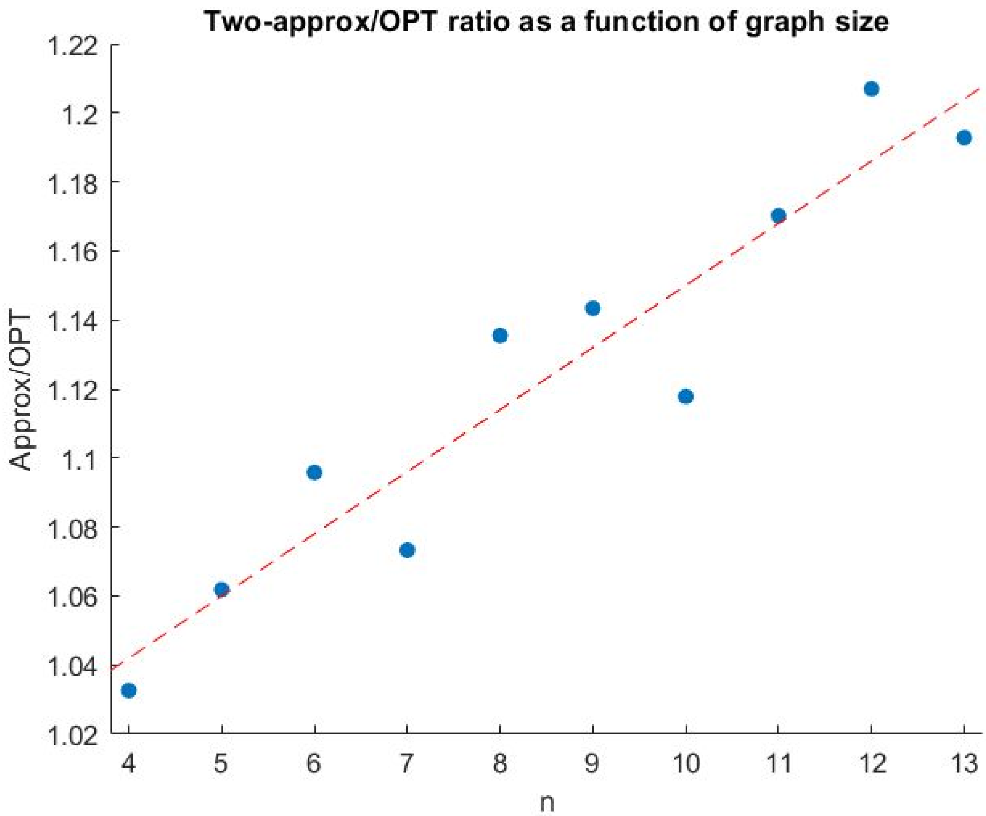
Step 4: Take shortcuts.



Extra: Java Implementation and analysis for 2 approximation

Implementation: https://github.com/alonmizrahi/tsp-2-approx

To analyze the 2-approximation algorithm and present practical results, for every graph size $n \in \{4, ..., 13\}$ we drew a random complete graph $G = (V, E)$ where $V \subseteq \mathbb{R}^2$, and then calculated the weight of every $u, v \in V$.
For every graph G, we calculated its 2 approximation cycle T^* as well as its minimum weight Hamiltonian cycle OPT (testing every possible permutation of vertices), and stored $d(T^*) / OPT^*$. We ran 16 tests for every $n \in \{4, ..., 13\}$ and averaged the results.
The results:



As we can see, the results seem to act like a straight line for $n \in \{4, ..., 13\}$, although we know it won't keep acting this way as we already proved $d(T^*) \leq 2 \cdot OPT^*$.
Therefore we can conclude that for higher n values, the 2 approximation bound might get somewhat tight.