

Advanced topics on Algorithms

Luciano Gualà

www.mat.uniroma2.it/~guala/

Approximation algorithms

Episode II

minimum Steiner Tree problem

minimum Steiner Tree problem

Input:

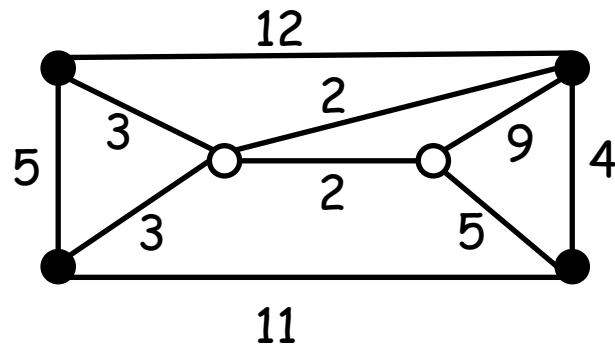
- undirected graph $G=(V,E)$ with non-negative edge costs
- subset of *required* vertices $R \subseteq V$; $V-R$ are called *Steiner* vertices

Feasible solution:

a tree T containing all the required vertices and any subset of Steiner ones

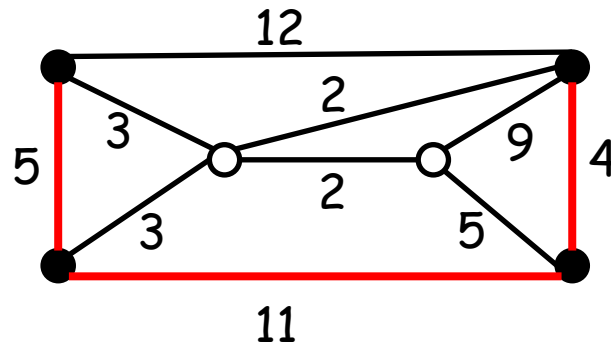
measure (min):

cost of T : $\sum_{e \in E(T)} c(e)$



● : required vertices

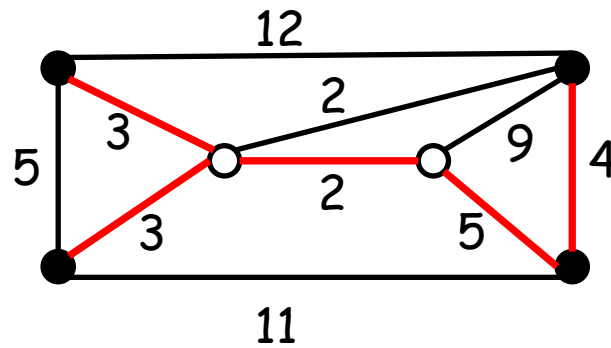
○ : Steiner vertices



a Steiner tree of cost 20

● : required vertices

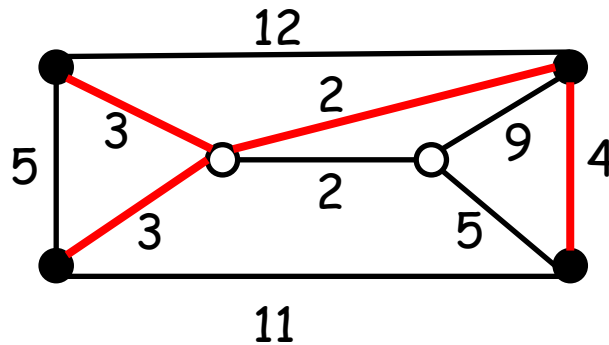
○ : Steiner vertices



a better Steiner tree of cost 17

● : required vertices

○ : Steiner vertices



a better Steiner tree of cost 12

special case: $R=V$

● : required vertices

○ : Steiner vertices

- Minimum Spanning Tree (MST) problem
- poly-time solvable

minimum Steiner Tree problem

Input:

- undirected graph $G=(V,E)$ with non-negative edge costs
- subset of *required* vertices $R \subseteq V$; $V-R$ are called *Steiner* vertices

Feasible solution:

a tree T containing all the required vertices and any subset of Steiner ones

measure (min):

$$\text{cost of } T : \sum_{e \in E(T)} c(e)$$

metric Steiner tree problem:

- G is complete, and
- edge costs satisfy the *triangle inequality*
for every u,v,w : $c(u,v) \leq c(u,w) + c(w,v)$

Theorem

There is an approximation factor preserving reduction from the Steiner tree problem to the metric Steiner tree problem.

proof

let I be an instance of the ST problem consisting of graph $G=(V,E)$ and required vertices R .


in poly-time

instance I' of metric ST problem:

- $G'=(V,E')$ complete; $c'(u,v)$ in G' = cost of any u - v shortest path in G
- $R'=R$

since for every $(u,v) \in E$, $c'(u,v) \leq c(u,v)$, $\text{OPT}(I') \leq \text{OPT}(I)$.

any steiner tree T' of I' can be converted in poly-time into a steiner tree T of I of at most the same cost:

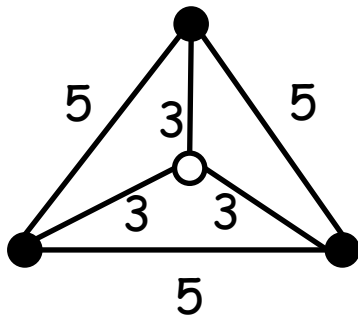
- replace each edge (u,v) of T' with the shortest path in G
- pick any spanning tree T of the obtained subgraph of G

$$\text{cost}(T) \leq \text{cost}(T')$$



Algorithm

output a Minimum Spanning Tree (MST) of the subgraph of G induced by R

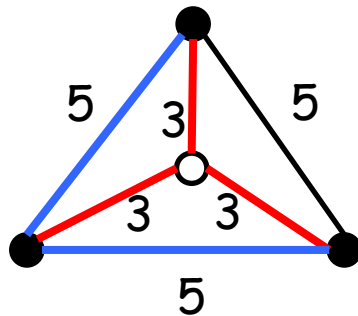


● : required vertices

○ : Steiner vertices

Algorithm

output a Minimum Spanning Tree (MST) of the subgraph of G induced by R



$OPT=9$

returned tree T has cost: 10

● : required vertices

○ : Steiner vertices

Theorem

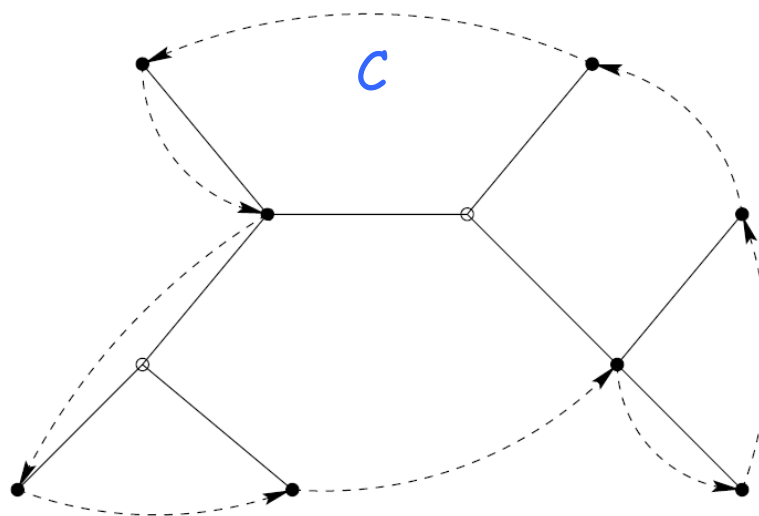
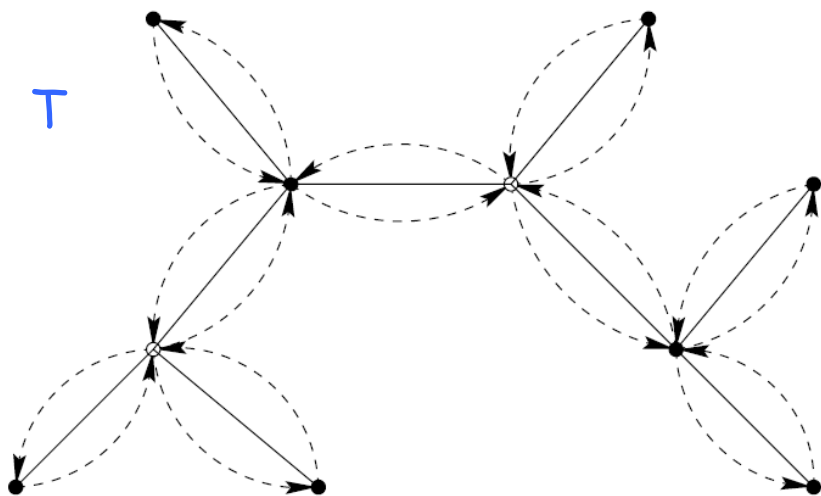
The algorithm is a 2-approximation algorithm for metric ST problem.

proof

let T be an optimal Steiner tree of cost OPT , and M the MST on R .

double the edges of T obtaining an Eulerian graph of cost $2\ OPT$

consider an Eulerian tour of cost $2\ OPT$



obtain a Hamiltonian cycle C on R by traversing the Eulerian tour and "shortcutting" Steiner vertices and previously visited vertices of R

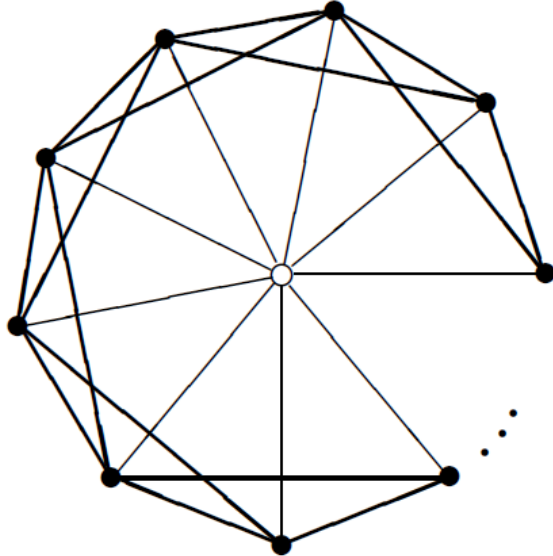
by triangle inequality: $\text{cost}(C) \leq 2\ OPT$

Since C is a spanning subgraph of $G[R]$: $\text{cost}(M) \leq \text{cost}(C)$



tight example

$n+1$
vertices



- edges incident to the Steiner vertex have cost 1
- all the other edges have cost 2

returned solution has
cost $2(n-1)$

$\text{OPT} = n$

Steiner Tree: state of the art

2	[Takahashi & Matsuyama, J.of Math. Jap, 1980]
$11/6 = 1.834$	[Zelikovsky, Algorithmica 93]
1.746	[Berman & Ramaiyer, SODA 92]
$1 + \ln 2 + \varepsilon = 1.693$	[Zelikovsky, Tech. Rep. 96]
$5/3 + \varepsilon = 1.667$	[Promel & Steger, STACS 96]
1.644	[Karpinski & Zelikovsky, JOCO 97]
1.598	[Hougardy & Promel, SODA 99]
$1 + (\ln 3)/2 + \varepsilon = 1.55$	[Robins & Zelikovsky, SODA 2000]
$\ln 4 + \varepsilon = 1.39$	[Byrka et al., STOC 2010]

Traveling Salesman Problem (TSP)

Traveling Salesman Problem

Input:

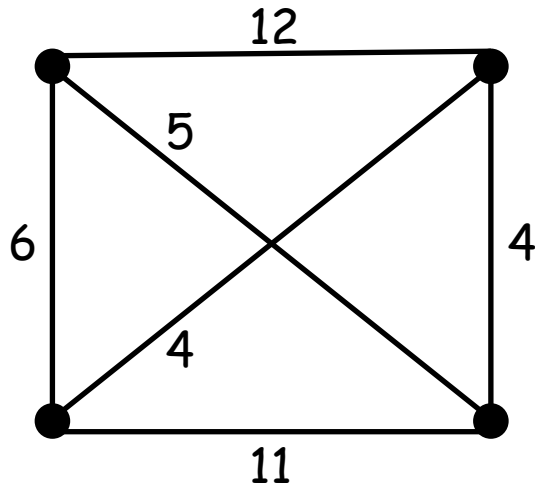
undirected complete graph $G=(V,E)$ with non-negative edge costs

Feasible solution:

a cycle C visiting every vertex exactly once

measure (min):

cost of C : $\sum_{e \in E(C)} c(e)$



Traveling Salesman Problem

Input:

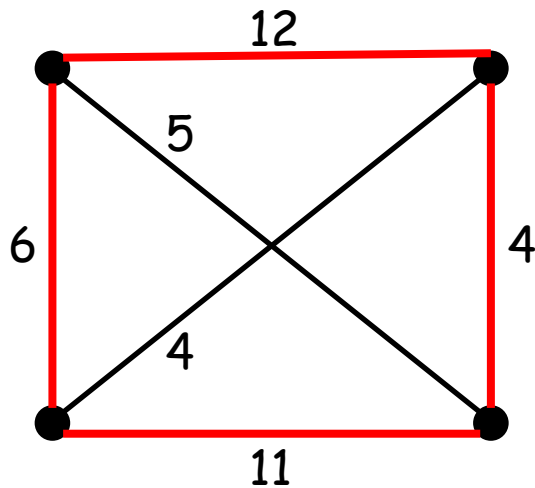
undirected complete graph $G=(V,E)$ with non-negative edge costs

Feasible solution:

a cycle C visiting every vertex exactly once

measure (min):

cost of C : $\sum_{e \in E(C)} c(e)$



a tour of cost 33

Traveling Salesman Problem

Input:

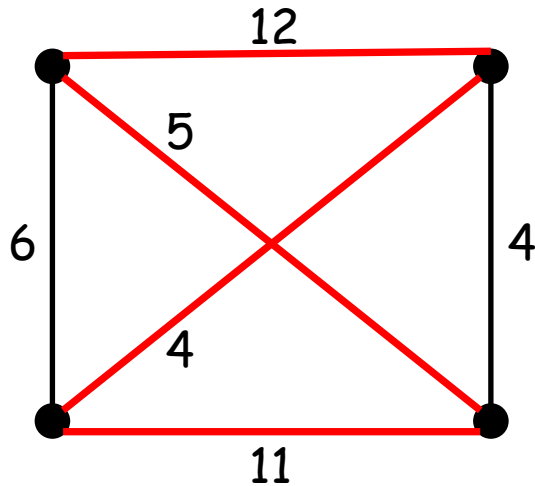
undirected complete graph $G=(V,E)$ with non-negative edge costs

Feasible solution:

a cycle C visiting every vertex exactly once

measure (min):

cost of C : $\sum_{e \in E(C)} c(e)$



a better tour of cost 32

Traveling Salesman Problem

Input:

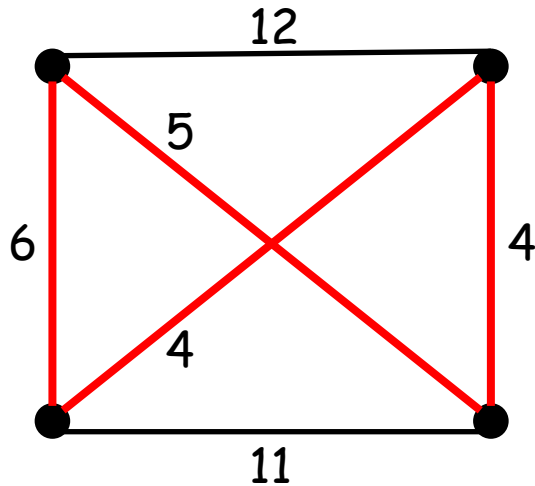
undirected complete graph $G=(V,E)$ with non-negative edge costs

Feasible solution:

a cycle C visiting every vertex exactly once

measure (min):

cost of C : $\sum_{e \in E(C)} c(e)$



a better tour of cost 19

Traveling Salesman Problem

Input:

undirected complete graph $G=(V,E)$ with non-negative edge costs

Feasible solution:

a cycle C visiting every vertex exactly once

measure (min):

cost of C : $\sum_{e \in E(C)} c(e)$

metric TSP:

edge costs satisfy the *triangle inequality*

for every u,v,w : $c(u,v) \leq c(u,w) + c(w,v)$

Theorem

For any polynomial time computable function $\alpha(n)$, TSP cannot be approximated within a factor of $\alpha(n)$, unless $P=NP$.

proof

by contradiction: let A be a $\alpha(n)$ -apx algorithm.

We use A to decide Hamiltonian cycle.

Let G be an instance of the Hamiltonian cycle. Define G' :

- $G'=(V,E')$ complete;
- $c(u,v)=1$ if $(u,v)\in E(G)$; $c(u,v)=n\alpha(n)$ otherwise

Clearly:

- if G has a Hamiltonian cycle, then optimal TSP tour in G' costs n
- if G does not have a Hamiltonian cycle, then optimal TSP tour is of cost $> n\alpha(n)$



G has an Hamiltonian cycle iff A returns a tour of cost n



Algorithm (metric TSP – factor 2)

1. Find an MST T of G
2. Double every edge of T to obtain an Eulerian graph
3. Find an Eulerian tour τ on this graph
4. Output the tour that visits vertices of G in the order of their first appearance in τ . Let C be this tour.

Theorem

The above algorithm is a 2-approximation algorithm for metric TSP.

proof

removing an edge from an optimal TSP tour gives us a spanning tree of G

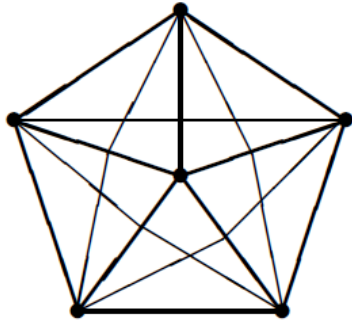
Thus: $\text{cost}(T) \leq \text{OPT}$

We have:

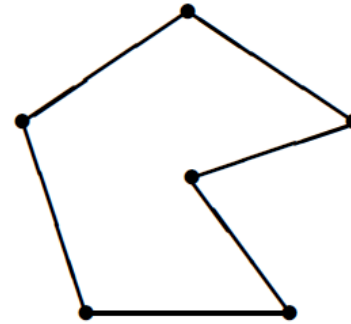
$$\text{cost}(C) \leq \text{cost}(\tau) = 2\text{cost}(T) \leq 2 \text{ OPT}$$



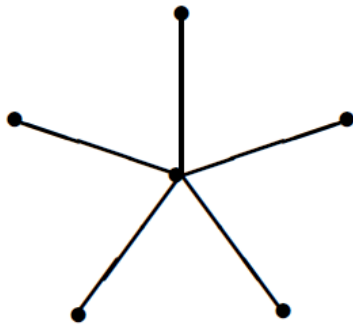
tight example



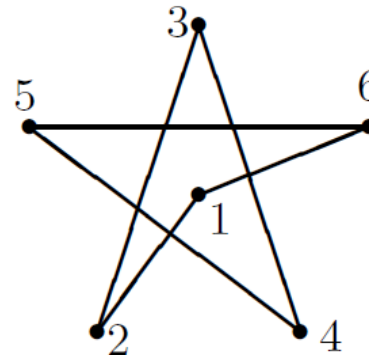
- n vertices
- thick edges have cost 1 (star+($n-1$)-cycle)
- all the other edges have cost 2



optimal tour of cost $OPT=n$



feasible MST



returned tour of cost $2n-2$
(for the feasible specified order)

idea: find a cheaper Eulerian subgraph/tour

recall:

- a graph is Eulerian iff all vertices have even degree
- in every undirected graph, the number of odd-degree vertices is even

Christofides, 1976

Algorithm (metric TSP – factor $3/2$)

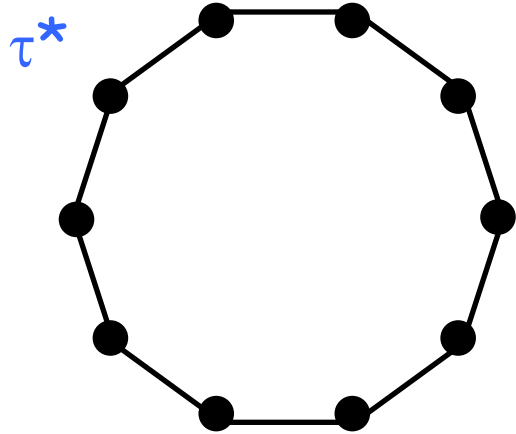
1. Find an MST T of G
2. Compute a minimum cost perfect matching, M , on the set V' of odd-degree vertices of T . Add M to T and obtain an Eulerian graph
3. Find an Eulerian tour τ on this graph
4. Output the tour that visits vertices of G in the order of their first appearance in τ . Let C be this tour.

Lemma

Let $V' \subseteq V$, such that $|V'|$ is even, and let M be a minimum cost perfect matching on V' . Then, $\text{cost}(M) \leq \text{OPT}/2$.

proof

let τ^* be an optimal TSP of cost OPT .

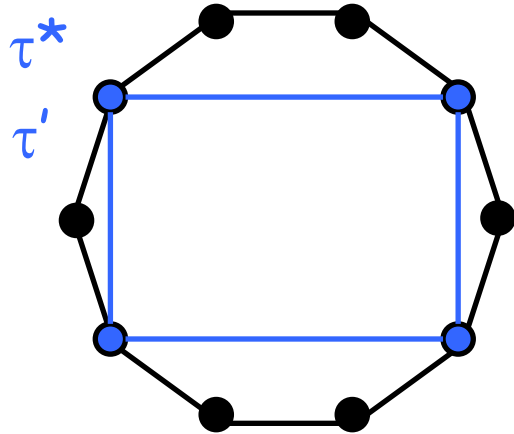


Lemma

Let $V' \subseteq V$, such that $|V'|$ is even, and let M be a minimum cost perfect matching on V' . Then, $\text{cost}(M) \leq \text{OPT}/2$.

proof

let τ^* be an optimal TSP of cost OPT .



$$\text{cost}(\tau') \leq \text{cost}(\tau^*)$$

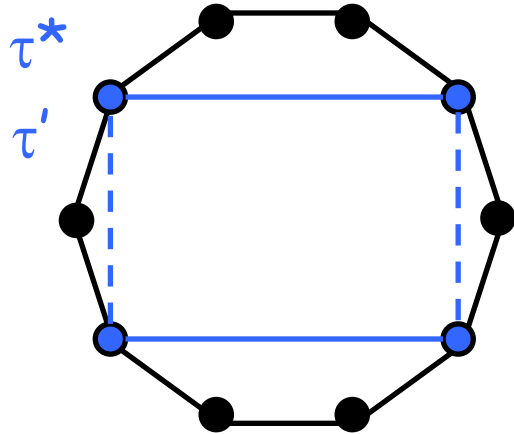
let τ' be the tour on V' obtained by shortcutting τ^* .

Lemma

Let $V' \subseteq V$, such that $|V'|$ is even, and let M be a minimum cost perfect matching on V' . Then, $\text{cost}(M) \leq \text{OPT}/2$.

proof

let τ^* be an optimal TSP of cost OPT .



$$\text{cost}(\tau') \leq \text{cost}(\tau^*)$$

let τ' be the tour on V' obtained by shortcutting τ^* .

τ' is the union of 2 perfect matching on V' , say M_1 and M_2 .

$$\text{cost}(M) \leq \min\{\text{cost}(M_1), \text{cost}(M_2)\} \leq \frac{1}{2} \text{cost}(\tau') \leq \frac{1}{2} \text{OPT}$$



Theorem

Christofides's algorithm is a $3/2$ -approximation algorithm for metric TSP.

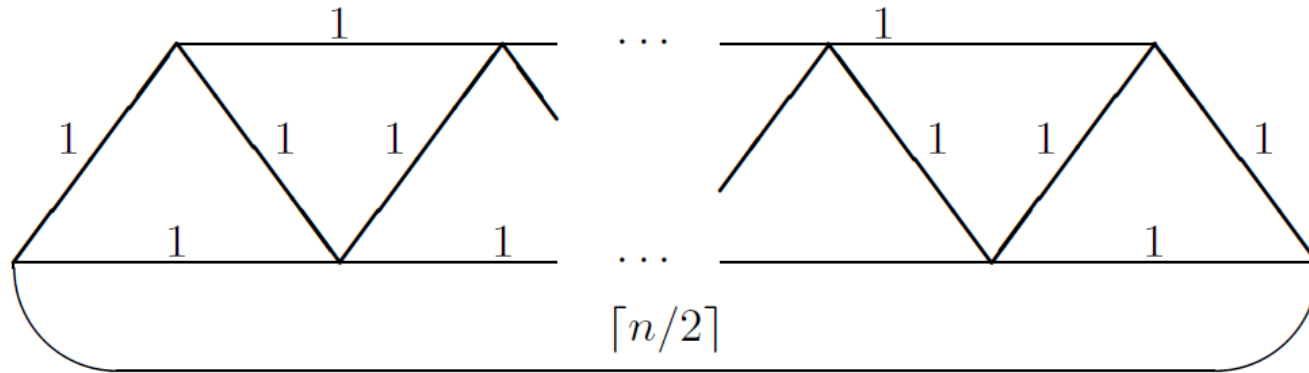
proof

We have:

$$\text{cost}(C) \leq \text{cost}(\tau) = \text{cost}(T) + \text{cost}(M) \leq \text{OPT} + \frac{1}{2} \text{OPT} \leq \frac{3}{2} \text{OPT}$$



tight example



- n vertices with n odd
- feasible MST: a path of $n-1$ edges
- matching: a single edge of cost $\lceil n/2 \rceil$

$$\text{OPT} = n$$

returned tour of cost $n-1 + \lceil n/2 \rceil$

TSP: state of the art

3/2 [Christofides, 1976]

STOC 2021

1 / 93 | — 100% + | □ ◇

A (Slightly) Improved Approximation Algorithm for Metric TSP

Anna R. Karlin*, Nathan Klein[†], and Shayan Oveis Gharan[‡]

University of Washington

March 16, 2022

Abstract

For some $\epsilon > 10^{-36}$ we give a randomized $3/2 - \epsilon$ approximation algorithm for metric TSP.