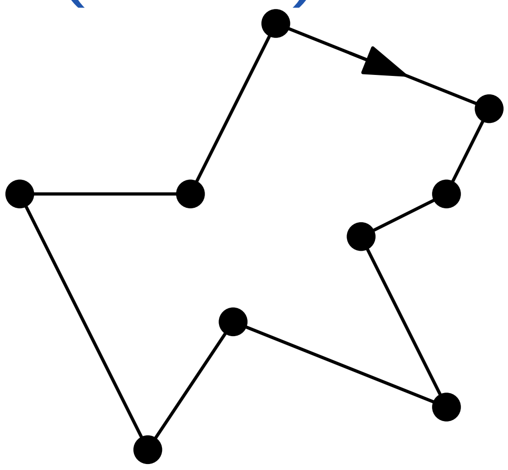
Advanced Algorithms for Geo-Information Systems

WiSe 2024/25

Approximation Algorithms

Alexander Naumann
Institut für Geodäsie und Geoinformation
Universität Bonn

Travelling Salesperson Problem (TSP)

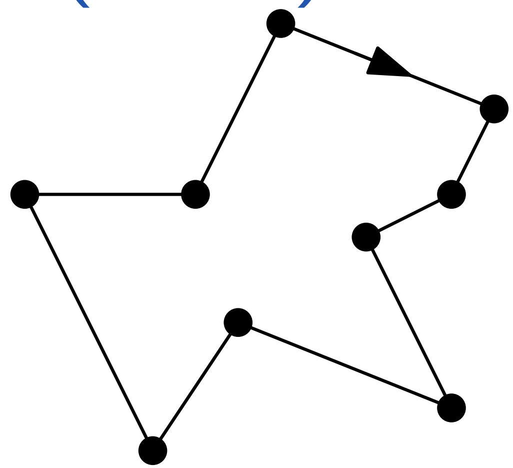


Input (optimization variant of TSP):
- n cities
- distances between the cities

Find a round trip of minimum total length visiting all cities

$\mathcal{NP}$-hard!

Travelling Salesperson Problem (TSP)



Input (optimization variant of TSP):

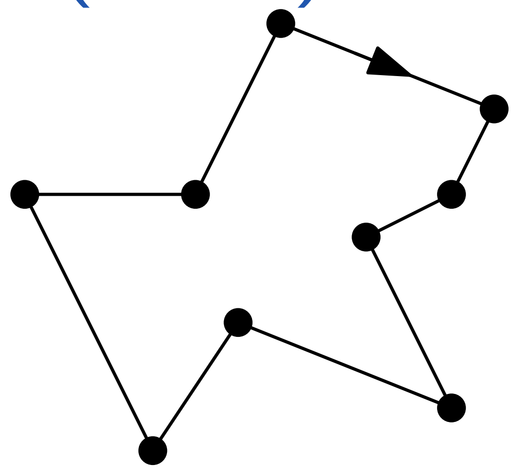
- n cities
- distances between the cities

Find a round trip of minimum total length visiting all cities

$\mathcal{NP}$-hard!

even if distance is a **metric**, which implies $D(a, c) \leq D(a, b) + D(b, c)$ for all $a, b, c \in S$

Travelling Salesperson Problem (TSP)



Input (optimization variant of TSP):

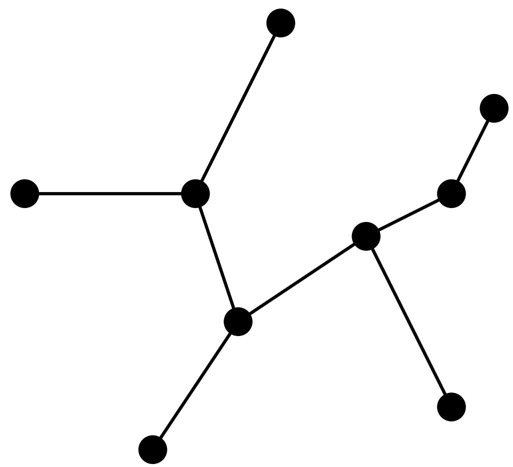
- n cities
- distances between the cities

Find a round trip of minimum total length visiting all cities

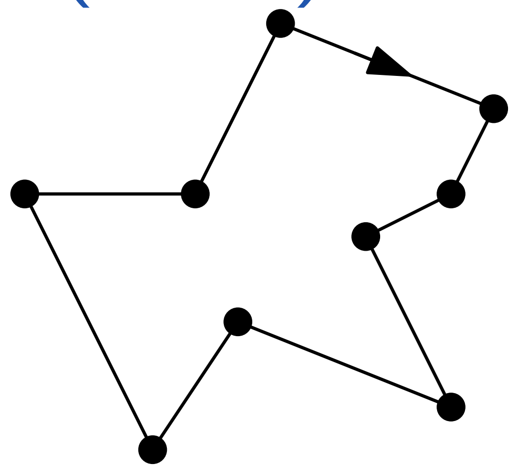
$\mathcal{NP}$-hard!

Algorithm for metric TSP:

- Compute minimum spanning tree MST of cities.



Travelling Salesperson Problem (TSP)



Input (optimization variant of TSP):

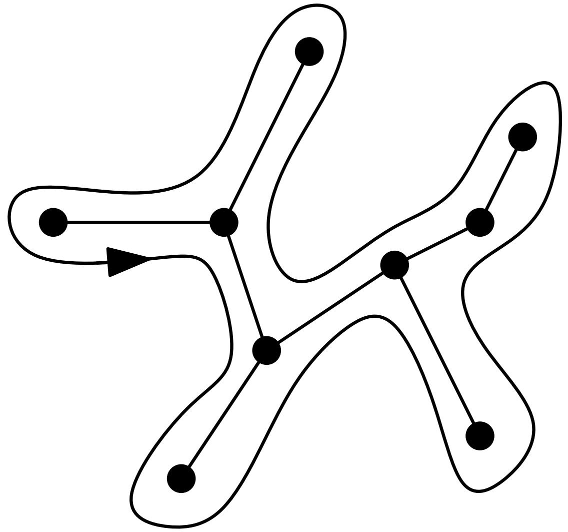
- n cities
- distances between the cities

Find a round trip of minimum total length visiting all cities

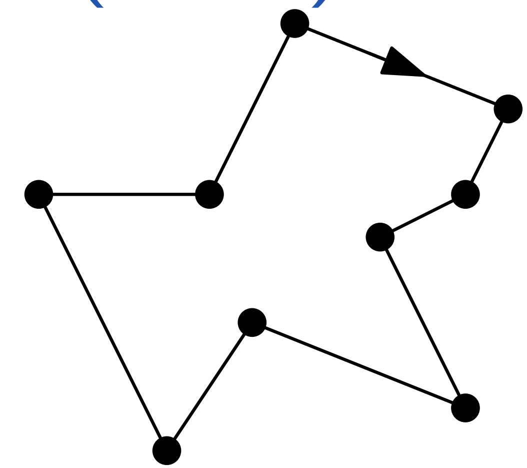
$\mathcal{NP}$-hard!

Algorithm for metric TSP:

- Compute minimum spanning tree MST of cities.
- Traverse each edge twice.



Travelling Salesperson Problem (TSP)



Input (optimization variant of TSP):

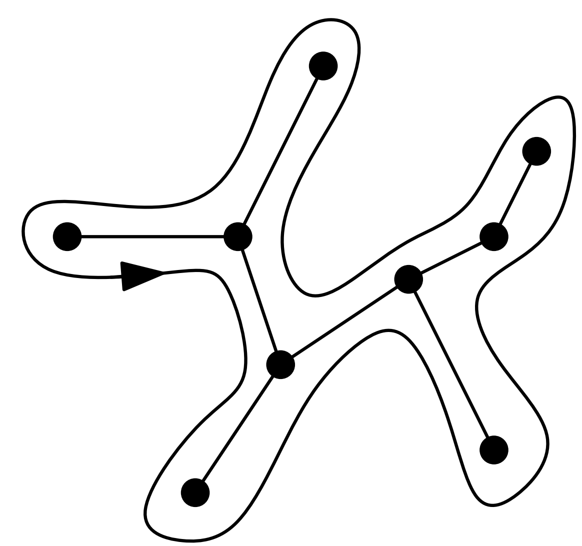
- n cities
- distances between the cities

Find a round trip of minimum total length visiting all cities

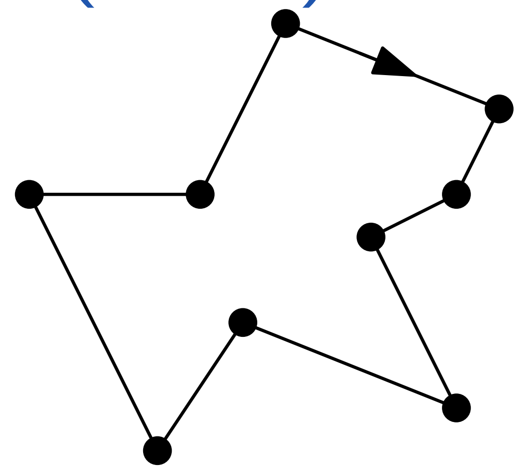
$\mathcal{NP}$-hard!

Algorithm for metric TSP:

- Compute minimum spanning tree MST of cities.
- Traverse each edge twice.
- Length of round trip: $L = 2 \cdot w(MST)$



Travelling Salesperson Problem (TSP)



Input (optimization variant of TSP):

- n cities
- distances between the cities

Find a round trip of minimum total length visiting all cities

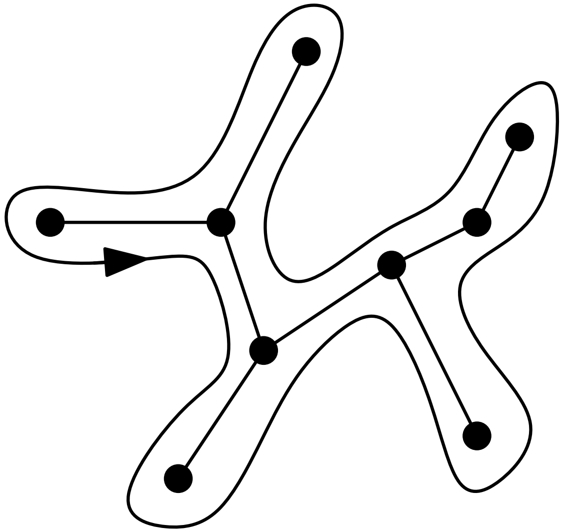
$\mathcal{NP}$-hard!

Algorithm for metric TSP:

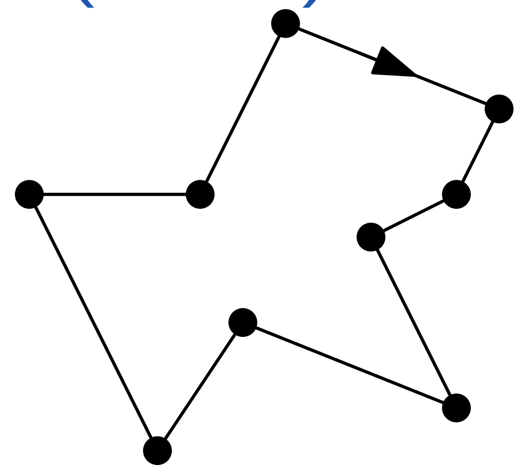
- Compute minimum spanning tree MST of cities.
- Traverse each edge twice.
- Length of round trip: $L = 2 \cdot w(MST)$

Observation: $w(MST) \leq opt$

Since a round trip (minus one edge) is also a spanning tree, but not necessarily optimal.



Travelling Salesperson Problem (TSP)



Input (optimization variant of TSP):

- n cities
- distances between the cities

Find a round trip of minimum total length visiting all cities

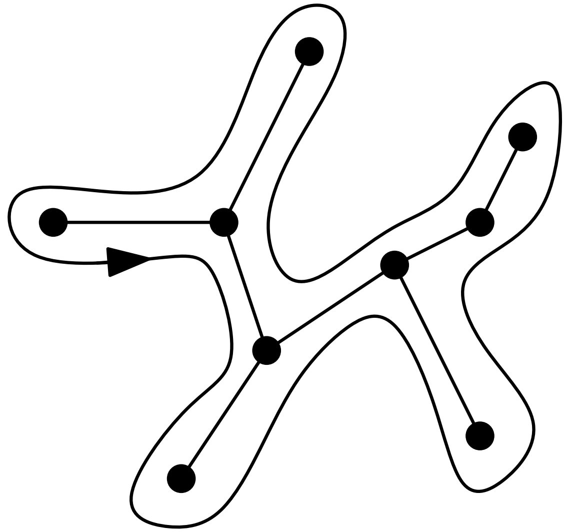
$\mathcal{NP}$-hard!

Algorithm for metric TSP:

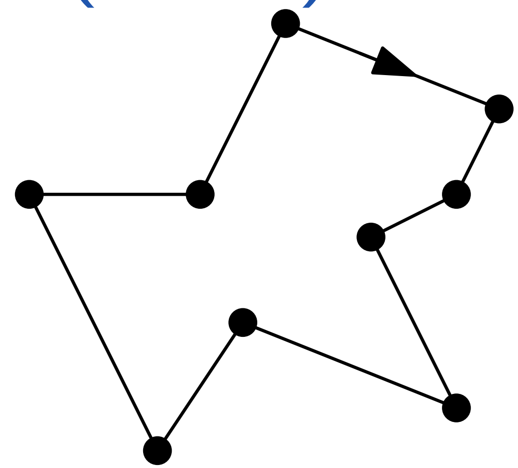
- Compute minimum spanning tree MST of cities.
- Traverse each edge twice.
- Length of round trip: $L = 2 \cdot w(MST) \leq 2 \cdot opt$

Observation: $w(MST) \leq opt$

Since a round trip (minus one edge) is also a spanning tree, but not necessarily optimal.



Travelling Salesperson Problem (TSP)



Input (optimization variant of TSP):
- n cities
- distances between the cities

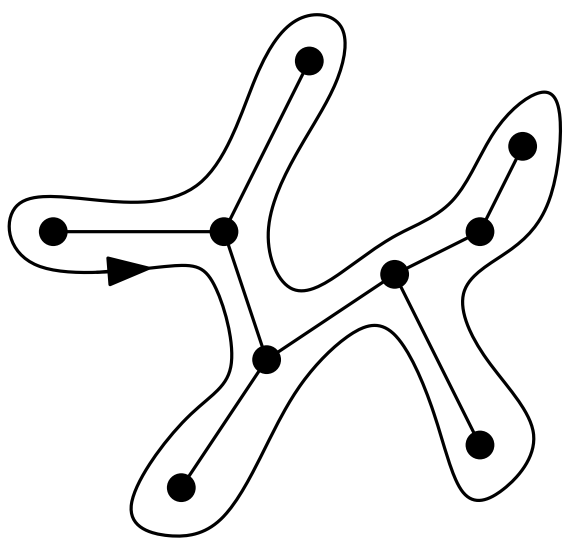
Find a round trip of minimum total length visiting all cities

$\mathcal{NP}$-hard!

Observation: $w(MST) \leq opt$

Since a round trip (minus one edge) is also a spanning tree, but not necessarily optimal.

Algorithm for metric TSP:
- Compute minimum spanning tree MST of cities.
- Traverse each edge twice.
- Length of round trip: $L = 2 \cdot w(MST) \leq 2 \cdot opt$



Length of solution is at most twice as long as optimal solution!

Approximation Algorithms

- An *approximation algorithm* yields a solution that is not worse than a certain factor α compared to an optimal solution.

Approximation Algorithms

- An *approximation algorithm* yields a solution that is not worse than a certain factor α compared to an optimal solution.
- Such a solution is termed an α-*approximation*.

Approximation Algorithms

- An *approximation algorithm* yields a solution that is not worse than a certain factor α compared to an optimal solution.
- Such a solution is termed an α-*approximation*.
- We do not need to compute an optimal solution to prove that the solution of our algorithm is an α-approximation!

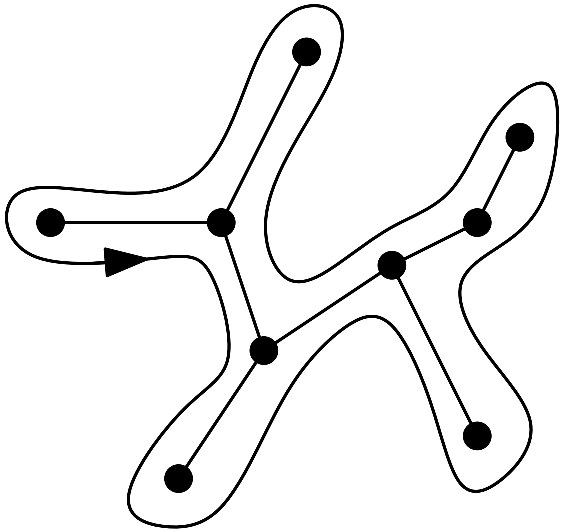
Approximation Algorithms

- An *approximation algorithm* yields a solution that is not worse than a certain factor α compared to an optimal solution.
- Such a solution is termed an α-*approximation*.
- We do not need to compute an optimal solution to prove that the solution of our algorithm is an α-approximation!
- A "2-approximation" sounds rather bad, but:
  a) $2 \cdot opt$ is an upper bound. Often the solution of an approximation algorithm is much better than the approximation guarantee.
  b) In contrast to approximation algorithms, heuristics do not provide any guarantee of quality!

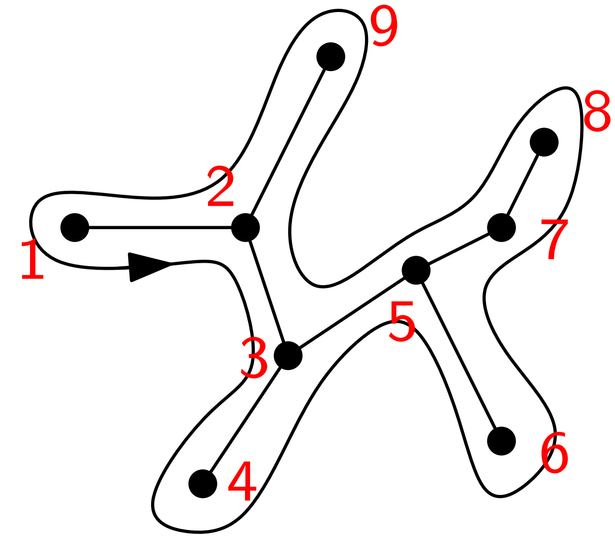
Travelling Salesperson Problem (TSP)

Algorithm for metric TSP:

- Compute minimum spanning tree MST of cities.
- Traverse each edge twice.
- Length of round trip: $L = 2 \cdot w(MST) \leq 2 \cdot opt$



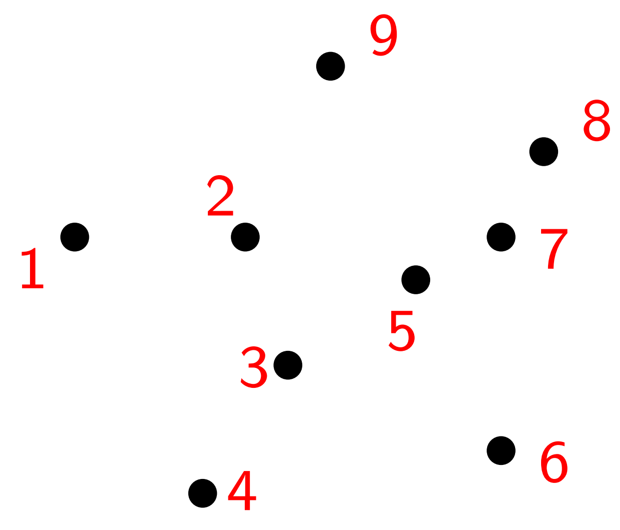
Travelling Salesperson Problem (TSP)



Algorithm for metric TSP:

- Compute minimum spanning tree MST of cities.
- Traverse each edge twice.
- Length of round trip: $L = 2 \cdot w(MST) \leq 2 \cdot opt$
- Number the cities in the order they occur on the round trip.

Travelling Salesperson Problem (TSP)



Algorithm for metric TSP:

- Compute minimum spanning tree MST of cities.
- Traverse each edge twice.
- Length of round trip: $L = 2 \cdot w(MST) \leq 2 \cdot opt$

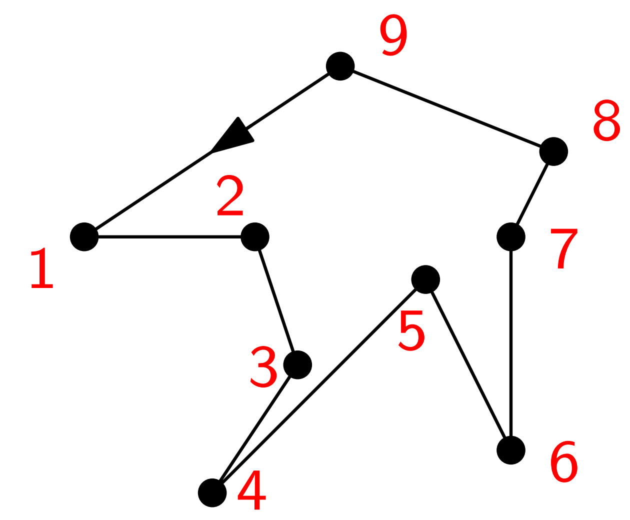
- Number the cities in the order they occur on the round trip.

- Visit the cities in that order – without visiting a city twice.

Travelling Salesperson Problem (TSP)

Algorithm for metric TSP:

- Compute minimum spanning tree MST of cities.
- Traverse each edge twice.
- Length of round trip: $L = 2 \cdot w(MST) \leq 2 \cdot opt$

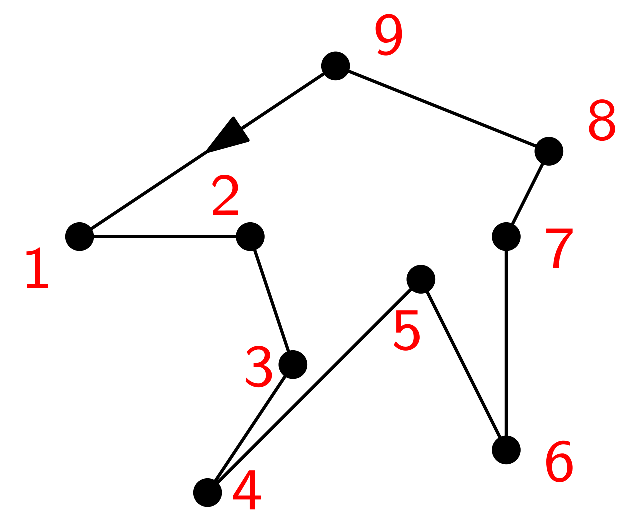


- Number the cities in the order they occur on the round trip.
- Visit the cities in that order – without visiting a city twice.
- Result:
  - a **simple** round trip
  - usually much better than $2 \cdot opt$ (while guarantee $\leq 2 \cdot opt$ still holds)

Travelling Salesperson Problem (TSP)

Algorithm for metric TSP:

- Compute minimum spanning tree MST of cities.
- Traverse each edge twice.
- Length of round trip: $L = 2 \cdot w(MST) \leq 2 \cdot opt$



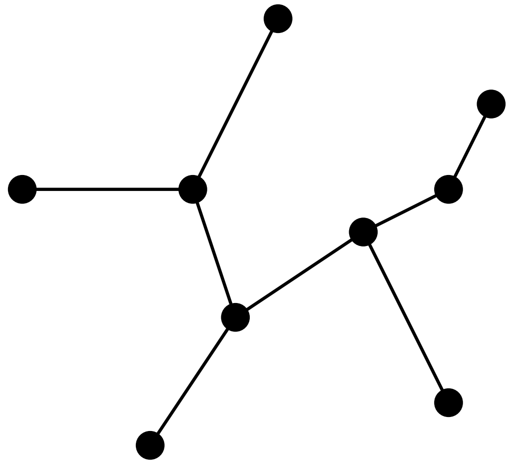
- Number the cities in the order they occur on the round trip.
- Visit the cities in that order – without visiting a city twice.
- Result:
  - a **simple** round trip
  - usually much better than $2 \cdot opt$ (while guarantee $\leq 2 \cdot opt$ still holds)

Is there an algorithm with approximation factor < 2 ?

The Algorithm of Christofides (1976)

Algorithm for metric TSP:

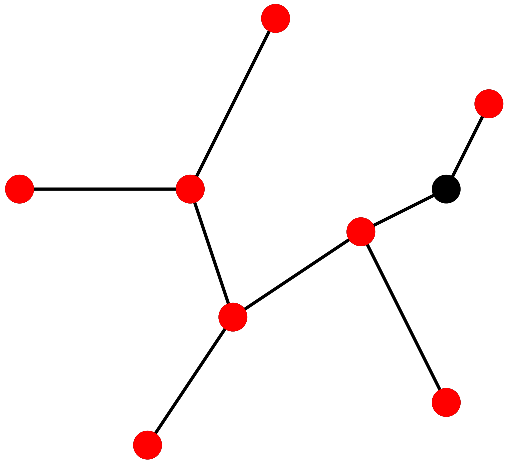
- Compute minimum spanning tree MST of cities.



The Algorithm of Christofides (1976)

Algorithm for metric TSP:

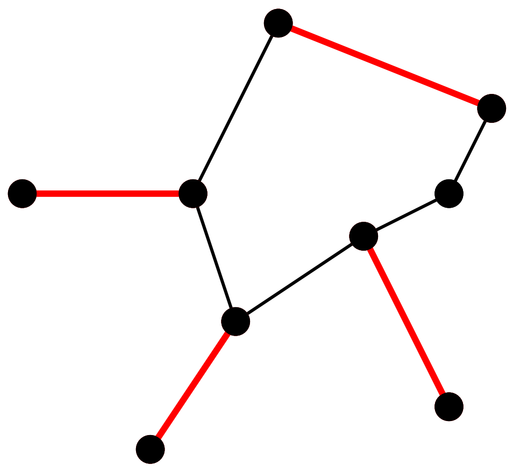
- Compute minimum spanning tree MST of cities.
- Let V be the set of cities with odd degree in MST.



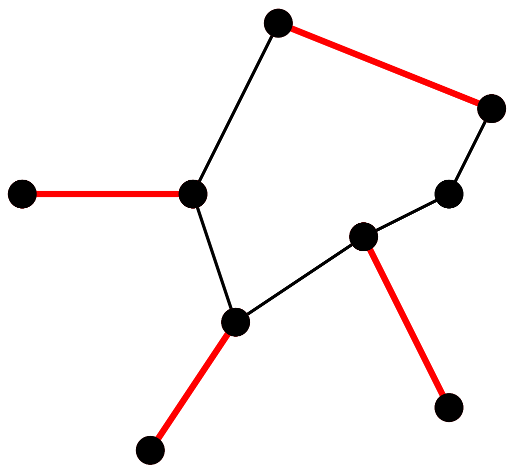
The Algorithm of Christofides (1976)

Algorithm for metric TSP:

- Compute minimum spanning tree MST of cities.
- Let V be the set of cities with odd degree in MST.
- Find in complete graph with node set V a shortest perfect matching.



The Algorithm of Christofides (1976)

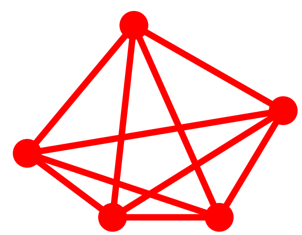


Algorithm for metric TSP:

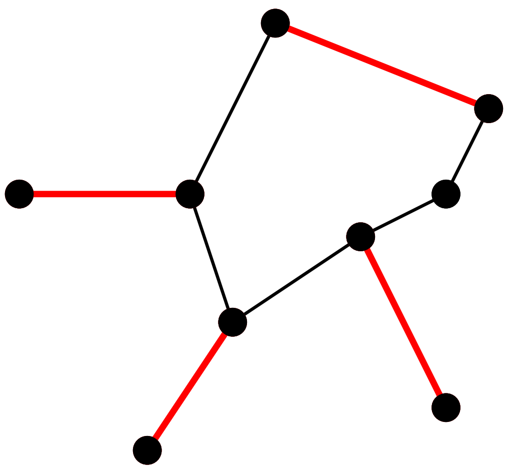
- Compute minimum spanning tree MST of cities.
- Let V be the set of cities with odd degree in MST.
- Find in complete graph with node set V a shortest perfect matching.

Terminology:

The **complete graph** for a given node set contains an edge connecting each two nodes.



The Algorithm of Christofides (1976)

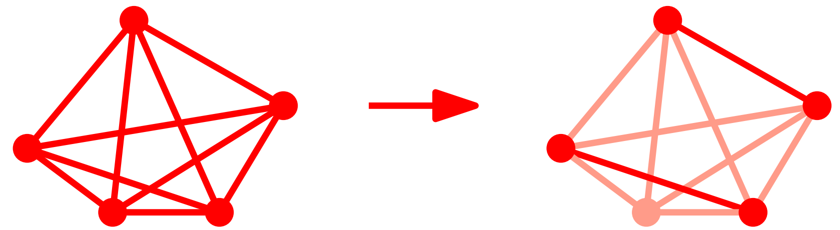


Algorithm for metric TSP:

- Compute minimum spanning tree MST of cities.
- Let V be the set of cities with odd degree in MST.
- Find in complete graph with node set V a shortest perfect matching.

Terminology:

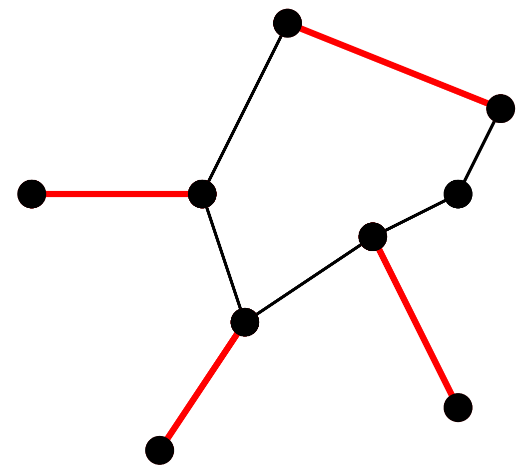
A **matching** of a graph $G = (V, E)$ is a subset of E that contains for each node in V **at most** one incident edge.



The Algorithm of Christofides (1976)

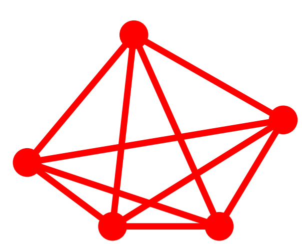
Algorithm for metric TSP:

- Compute minimum spanning tree MST of cities.
- Let V be the set of cities with odd degree in MST.
- Find in complete graph with node set V a shortest perfect matching.



Terminology:

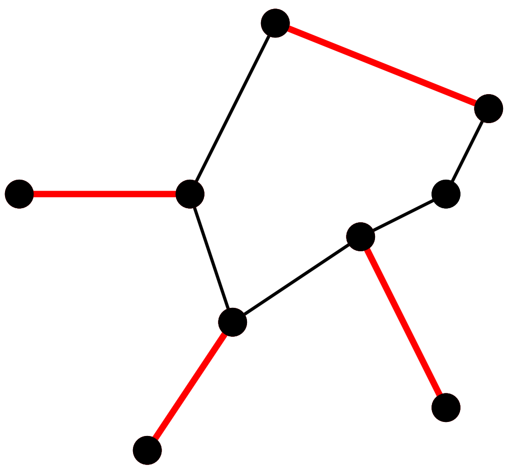
A **perfect matching** of a graph $G = (V, E)$ is a subset of E that contains for each node in V **exactly** one incident edge.



The Algorithm of Christofides (1976)

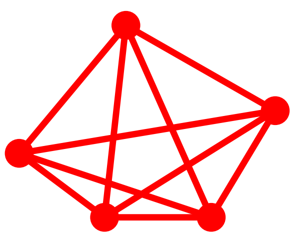
Algorithm for metric TSP:

- Compute minimum spanning tree MST of cities.
- Let V be the set of cities with odd degree in MST.
- Find in complete graph with node set V a shortest perfect matching.



Terminology:

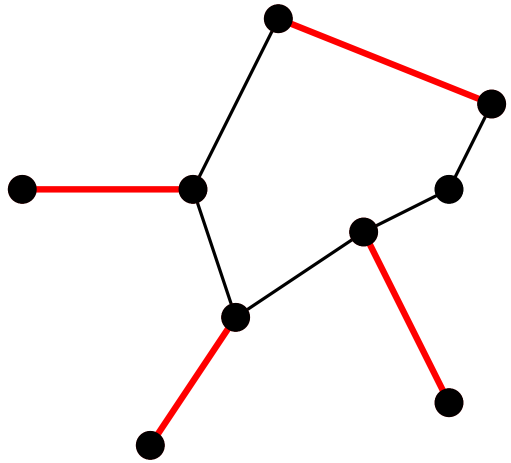
A **perfect matching** of a graph $G = (V, E)$ is a subset of E that contains for each node in V **exactly** one incident edge.



A complete graph $G = (V, E)$ has a perfect matching if and only if $|V|$ is even.

The Algorithm of Christofides (1976)

Algorithm for metric TSP:

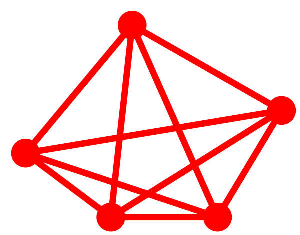


- Compute minimum spanning tree MST of cities.
- Let V be the set of cities with odd degree in MST.
- Find in complete graph with node set V a shortest perfect matching.

Given in our case, since $\sum_{v \in V} deg(v)$ is even.

Terminology:

A **perfect matching** of a graph $G = (V, E)$ is a subset of E that contains for each node in V **exactly** one incident edge.

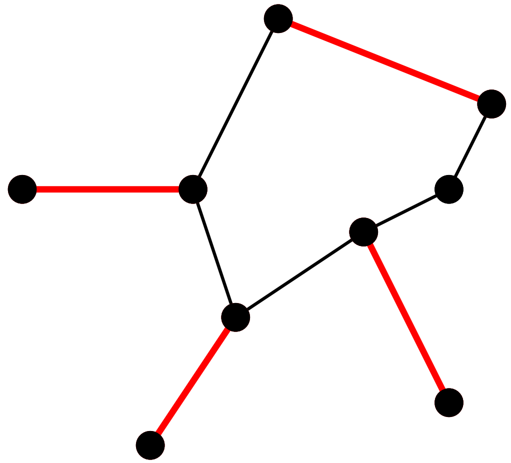


A complete graph $G = (V, E)$ has a perfect matching if and only if $|V|$ is even.

The Algorithm of Christofides (1976)

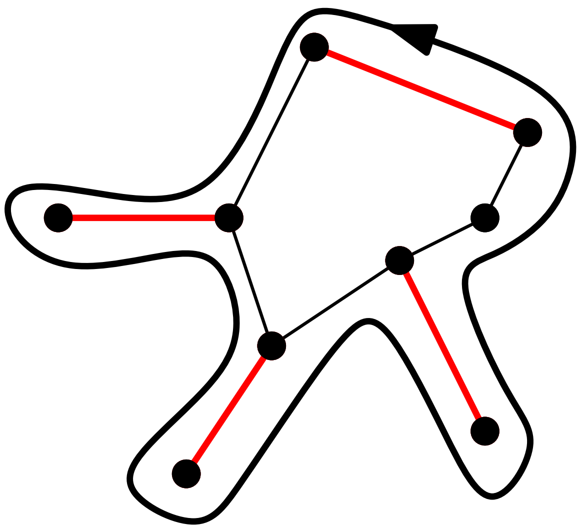
Algorithm for metric TSP:

- Compute minimum spanning tree MST of cities.
- Let V be the set of cities with odd degree in MST.
- Find in complete graph with node set V a shortest perfect matching.



can be computed in $O(|V|^3)$ time

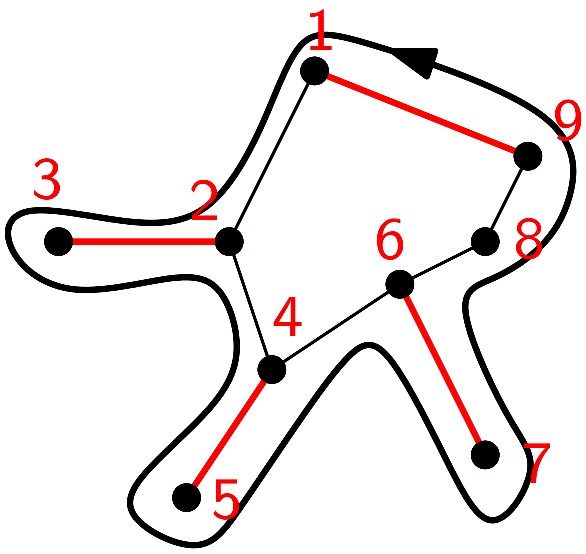
The Algorithm of Christofides (1976)



Algorithm for metric TSP:

- Compute minimum spanning tree MST of cities.
- Let V be the set of cities with odd degree in MST.
- Find in complete graph with node set V a shortest perfect matching.
- Find an Euler circuit in the multigraph that contains all MST edges and all edges of the matching.

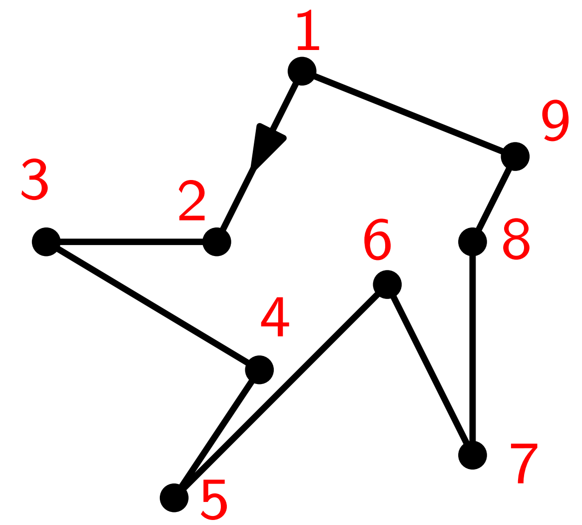
The Algorithm of Christofides (1976)



Algorithm for metric TSP:

- Compute minimum spanning tree MST of cities.
- Let V be the set of cities with odd degree in MST.
- Find in complete graph with node set V a shortest perfect matching.
- Find an Euler circuit in the multigraph that contains all MST edges and all edges of the matching.
- Number the nodes in the order of their first visits.

The Algorithm of Christofides (1976)



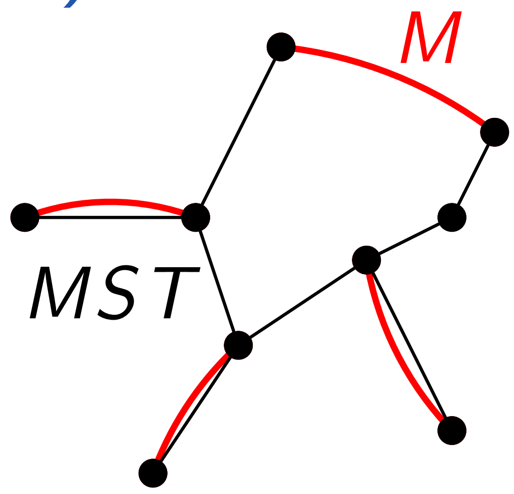
Algorithm for metric TSP:

- Compute minimum spanning tree MST of cities.
- Let V be the set of cities with odd degree in MST.
- Find in complete graph with node set V a shortest perfect matching.
- Find an Euler circuit in the multigraph that contains all MST edges and all edges of the matching.
- Number the nodes in the order of their first visits.
- Visit the nodes in the given order.

The Algorithm of Christofides (1976)

Theorem: The round trip R returned by the algorithm of Chr. is a 1.5-approximation.

The Algorithm of Christofides (1976)

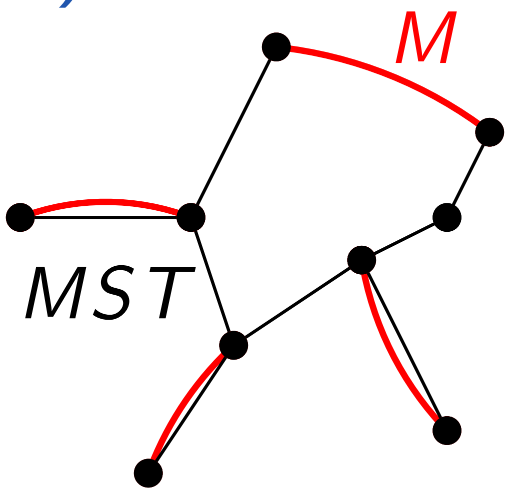


Theorem: The round trip R returned by the algorithm of Chr. is a 1.5-approximation.

Proof:

It holds $w(R) \leq w(MST) + w(M)$.

The Algorithm of Christofides (1976)



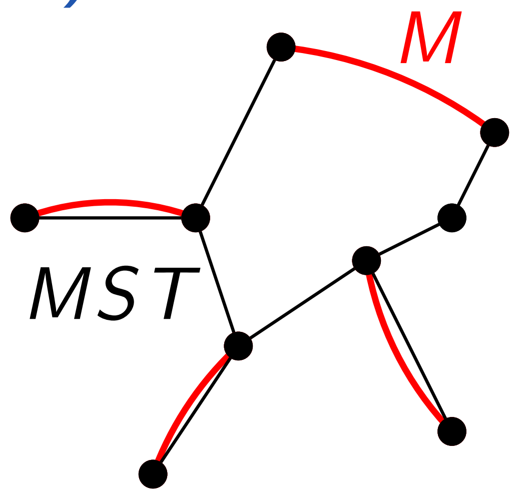
Theorem: The round trip R returned by the algorithm of Chr. is a 1.5-approximation.

Proof:

It holds $w(R) \leq w(MST) + w(M)$.

Furthermore, it holds $w(MST) \leq opt$.

The Algorithm of Christofides (1976)



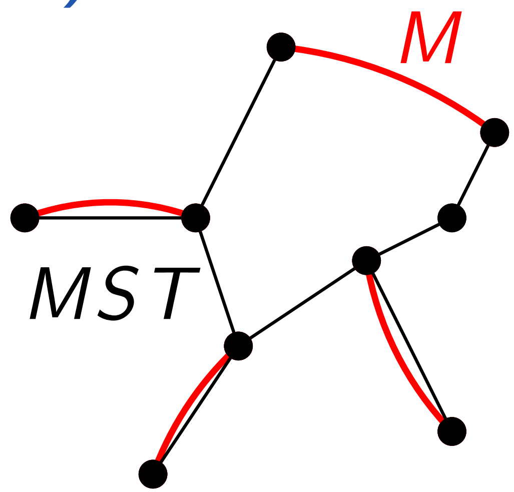
Theorem: The round trip R returned by the algorithm of Chr. is a 1.5-approximation.

Proof:

It holds $w(R) \leq w(MST) + w(M)$.

Furthermore, it holds $w(MST) \leq opt$. To show: $w(M) \leq \frac{1}{2} opt$

The Algorithm of Christofides (1976)



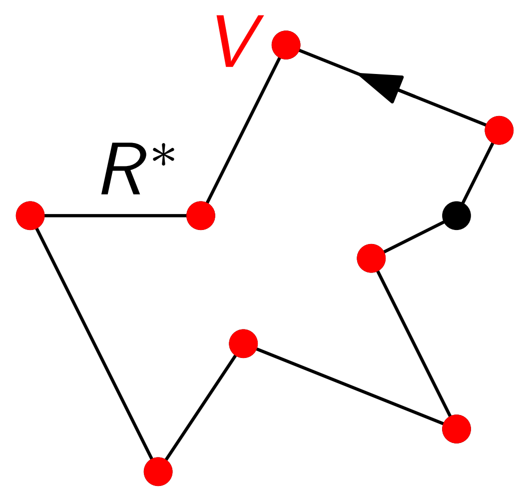
Theorem: The round trip R returned by the algorithm of Chr. is a 1.5-approximation.

Proof:

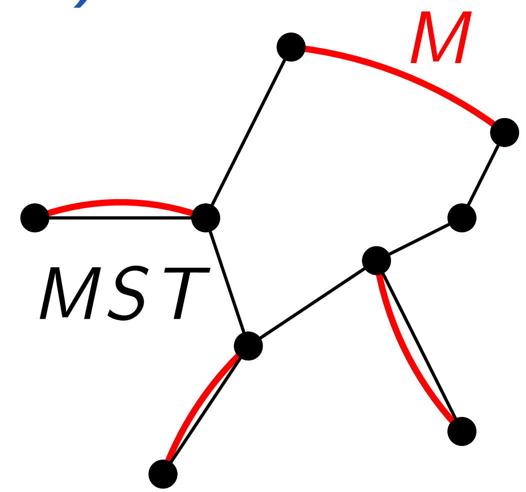
It holds $w(R) \leq w(MST) + w(M)$.

Furthermore, it holds $w(MST) \leq opt$. To show: $w(M) \leq \frac{1}{2} opt$

- Let R^* be an optimal round trip and V the set of nodes with odd degree in MST.



The Algorithm of Christofides (1976)



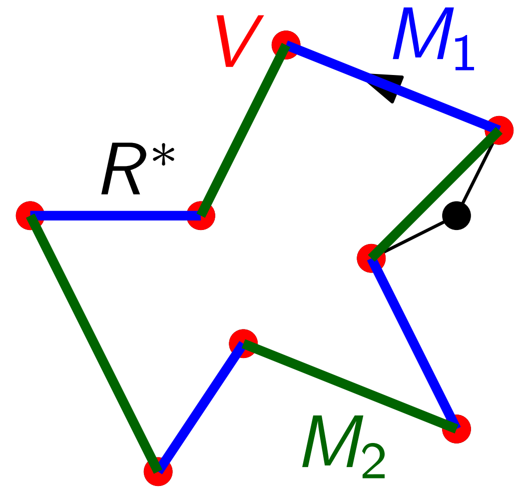
Theorem: The round trip R returned by the algorithm of Chr. is a 1.5-approximation.

Proof:

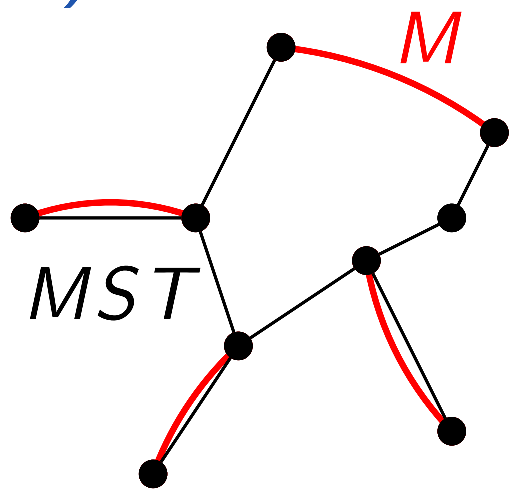
It holds $w(R) \leq w(MST) + w(M)$.

Furthermore, it holds $w(MST) \leq opt$. To show: $w(M) \leq \frac{1}{2} opt$

- Let R^* be an optimal round trip and V the set of nodes with odd degree in MST.
- R^* defines two perfect matchings M_1 and M_2 in the complete graph G for V.



The Algorithm of Christofides (1976)



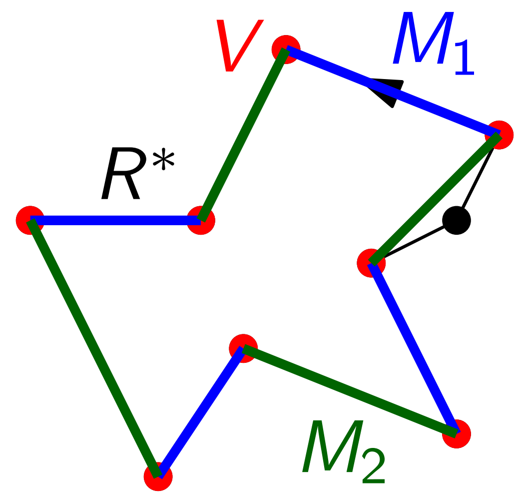
Theorem: The round trip R returned by the algorithm of Chr. is a 1.5-approximation.

Proof:

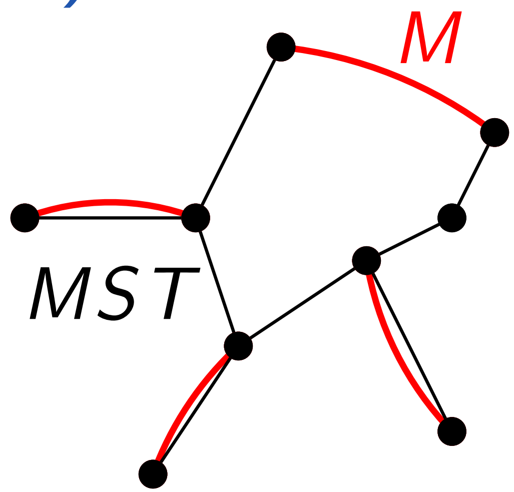
It holds $w(R) \leq w(MST) + w(M)$.

Furthermore, it holds $w(MST) \leq opt$. To show: $w(M) \leq \frac{1}{2}opt$

- Let R^* be an optimal round trip and V the set of nodes with odd degree in MST.
- R^* defines two perfect matchings M_1 and M_2 in the complete graph G for V.
- It holds $w(M_1) + w(M_2) \leq opt$.



The Algorithm of Christofides (1976)



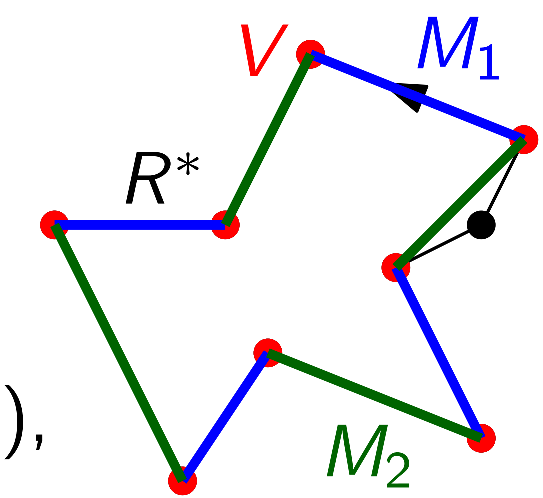
Theorem: The round trip R returned by the algorithm of Chr. is a 1.5-approximation.

Proof:

It holds $w(R) \leq w(MST) + w(M)$.

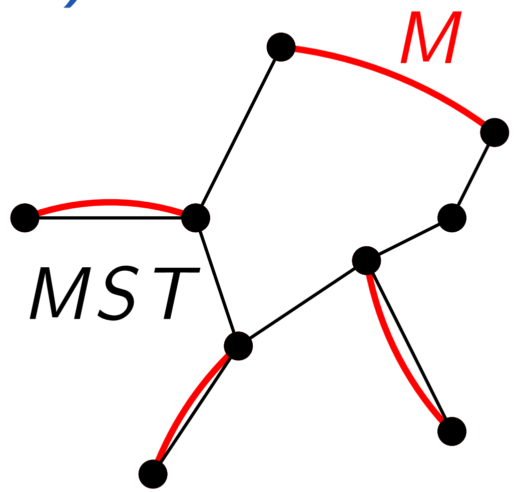
Furthermore, it holds $w(MST) \leq opt$. To show: $w(M) \leq \frac{1}{2} opt$

- Let R^* be an optimal round trip and V the set of nodes with odd degree in MST.
- R^* defines two perfect matchings M_1 and M_2 in the complete graph G for V.
- It holds $w(M_1) + w(M_2) \leq opt$.
- It holds $w(M) \leq w(M_1)$ and $w(M) \leq w(M_2)$, since M is a **shortest** perfect matching in G.



The Algorithm of Christofides (1976)

Theorem: The round trip R returned by the algorithm of Chr. is a 1.5-approximation.



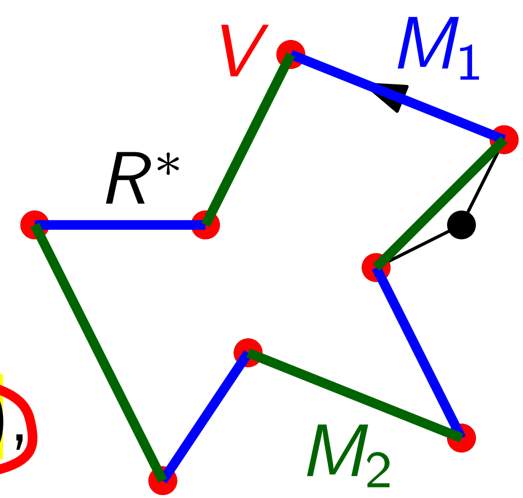
Proof:

It holds $w(R) \leq w(MST) + w(M)$.

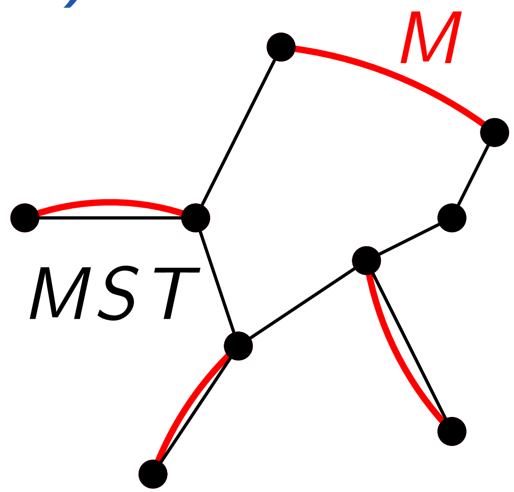
Furthermore, it holds $w(MST) \leq opt$. To show: $w(M) \leq \frac{1}{2} opt$

$2w(M) \leq opt$

- Let R^* be an optimal round trip and V the set of nodes with odd degree in MST.
- R^* defines two perfect matchings M_1 and M_2 in the complete graph G for V.
- It holds $w(M_1) + w(M_2) \leq opt$.
- It holds $w(M) \leq w(M_1)$ and $w(M) \leq w(M_2)$, since M is a **shortest** perfect matching in G.



The Algorithm of Christofides (1976)



Theorem: The round trip R returned by the algorithm of Chr. is a 1.5-approximation.

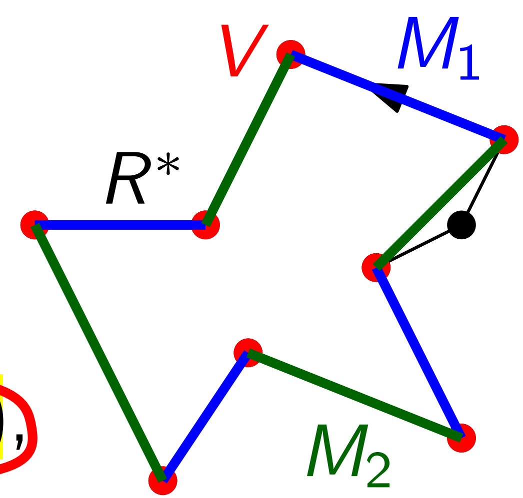
Proof:

It holds $w(R) \leq w(MST) + w(M)$.

Furthermore, it holds $w(MST) \leq opt$. To show: $w(M) \leq \frac{1}{2} opt$

$2w(M) \leq opt$

- Let R^* be an optimal round trip and V the set of nodes with odd degree in MST.
- R^* defines two perfect matchings M_1 and M_2 in the complete graph G for V.
- It holds $w(M_1) + w(M_2) \leq opt$.
- It holds $w(M) \leq w(M_1)$ and $w(M) \leq w(M_2)$, since M is a **shortest** perfect matching in G.



□

Summary

- Approximation algorithms are often used for $\mathcal{NP}$-hard optimization problems.
- Other than ILP-based methods, they are inexact but efficient.
- Other than heuristics, they provide a guarantee of quality.
- Designing approximation algorithms and proving their guarantees is a common algorithmic challenge.