§8 Approximation Algorithms

- Motivation and Classification
- *metric* Travelling Salesperson
  - *Double-Tree*: approximation ratio 2
  - *Christofides*: approximation ratio 1.5
- Knapsack
  - Strongly polyn.-time / Dynamic Programming
  - *Fully* Polynomial-Time Approximation *Scheme*
- Limits of Approximability

Motivation and Classification

Many important *decision* problems

- may not admit a *polynomial*-time solution
- arise from **optimization** problems

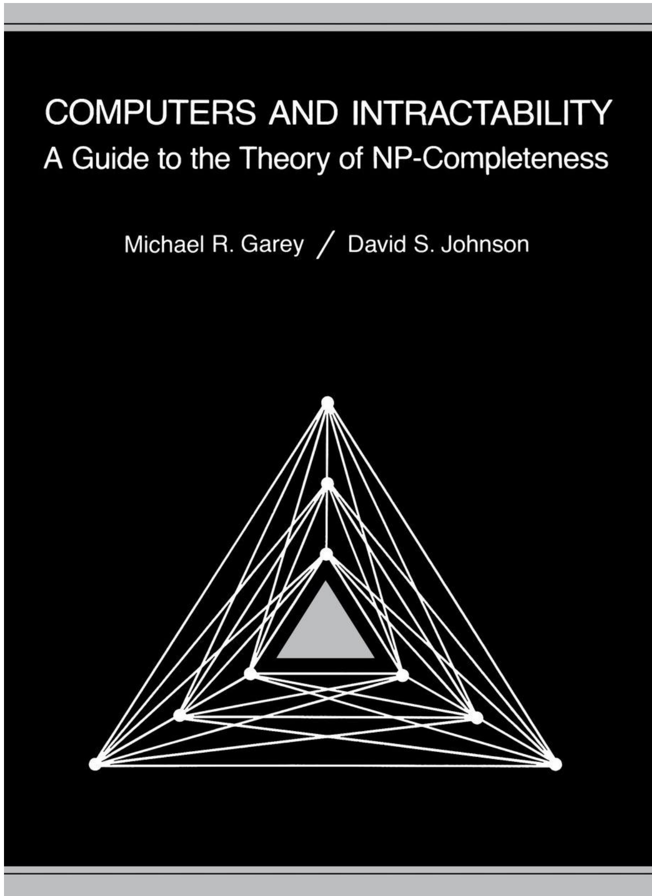
Approximation factor or ratio:

- C/C^* for *minimization* problems,
- C^*/C for *maximization* problems.

C^*=optimal, C=approxim. answer

exact *exponential*-time algorithms:

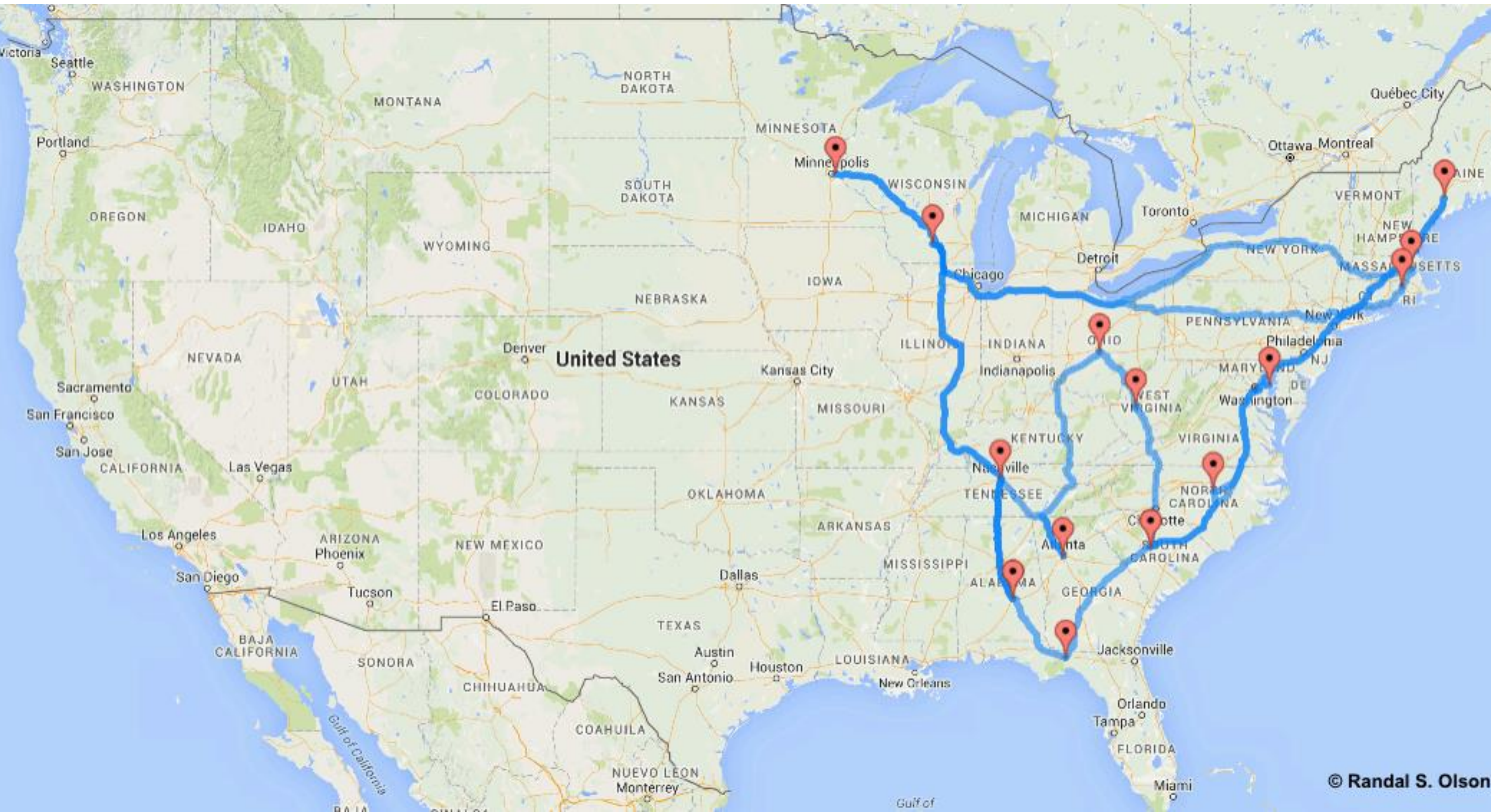
- $O(b^n)$ for "small" base $b>1$
- parameterized, *strong* polynom.



Travelling Salesperson Problem

HC = { ⟨G⟩ | G has Hamiltonian cycle}

TSP = { ⟨G,k⟩ | <u>weighted</u> graph G contains cycle of length $\leq k$ }



© Randal S. Olson

Approximating metric TSP

MTSP = { $\langle G, \underline{w}, k\rangle$ | G with metric edge weights $\underline{w}: V\times V \to \mathbb{N}$ admits a Hamiltonian circuit of weighted length $\leq k$ }

Input: $\underline{w}: V\times V\to\mathbb{N}$ edge weights symmetric and s.t. triangle inequality holds: $w(a,c) \leq w(a,b)+w(b,c)$.

Sought: Tour (permutation π of V) of least weight

(Decision problem **MTSP** is still $\mathcal{NP}$-complete.)

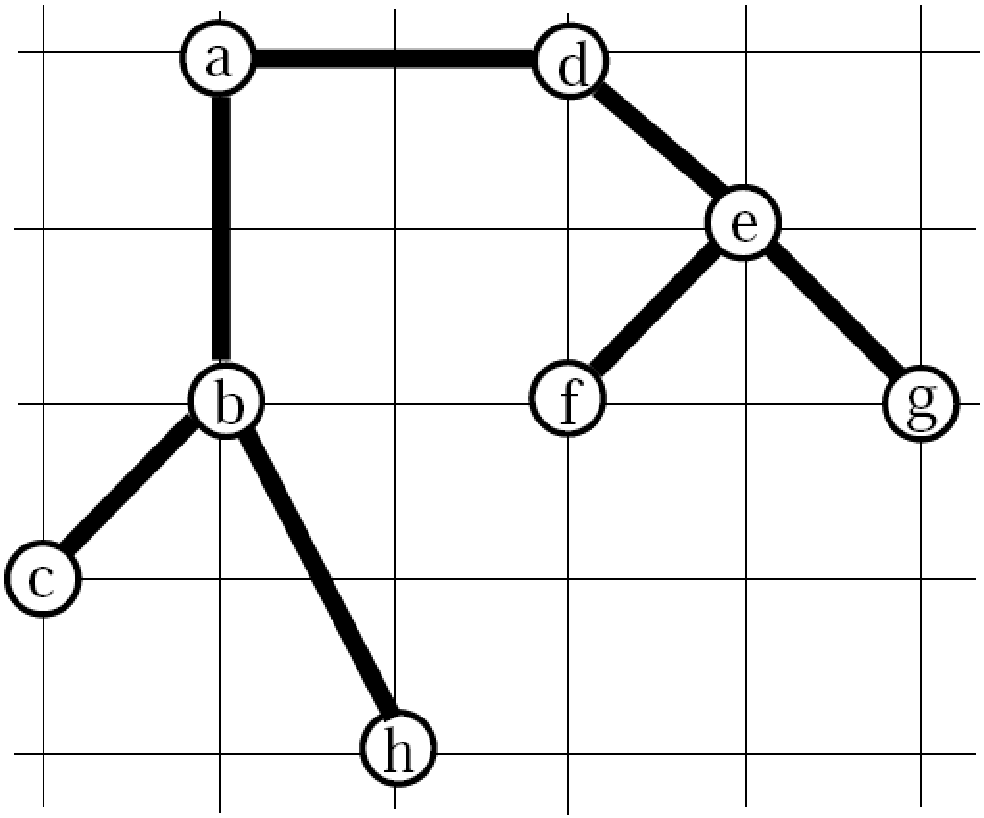
***Double-Tree* Algorithm:**

Polytime approximating **minMTSP** up to factor 2:

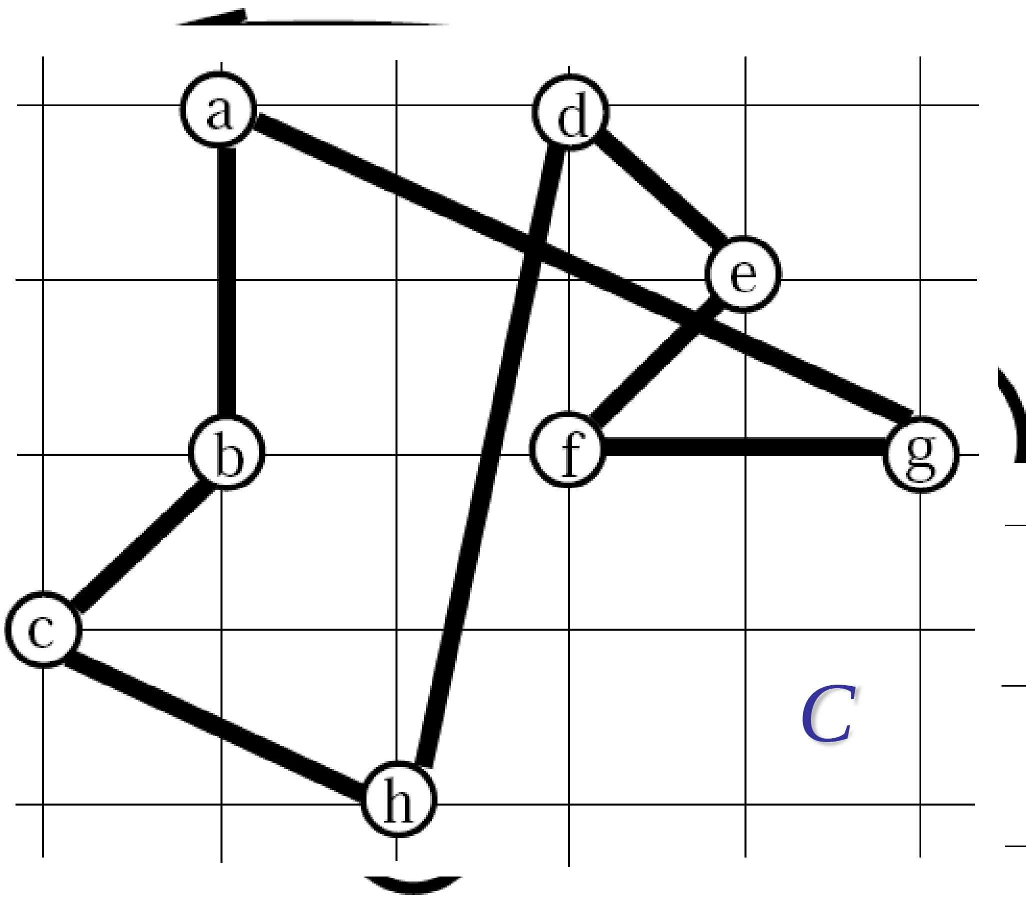
1. Compute minimum spanning tree T of (G,w).
2. Traverse T inorder depth-first

ETSP with $V\subseteq\mathbb{R}^d$, $w(\underline{a},\underline{b}) = \|\underline{a}-\underline{b}\|_2$ $\mathcal{NP}$-hard, but **in $\mathcal{NP}$?**

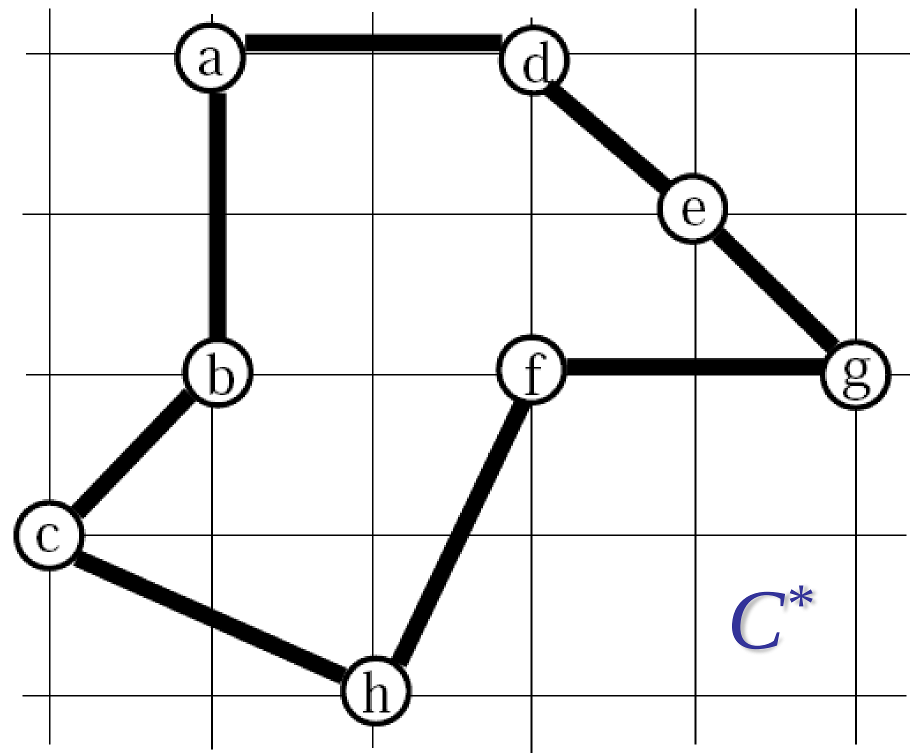
Example: 1. Compute MST $\boldsymbol{T}$



Example: 2. Traverse T *inorder*



output:
a,b,c,h,d,e,f,g,a



optimal: a,b,c,h,f,g,e,d,a

Proof of Approximation Ratio

w weights with 3-inequality, T is MST traversed inorder

Let F denote the sequence of edges traversed inorder, C the tour thus obtained, C^* (unknown) optimal tour.

For edges $e_1,\ldots e_k$ abbreviate $L(e_1,\ldots e_k) := w(e_1)+\ldots+w(e_k)$

(i) $L(T) \leq L(C^*)$, because removing any edge from C^* yields a spanning tree of cost $\leq L(C^*)$

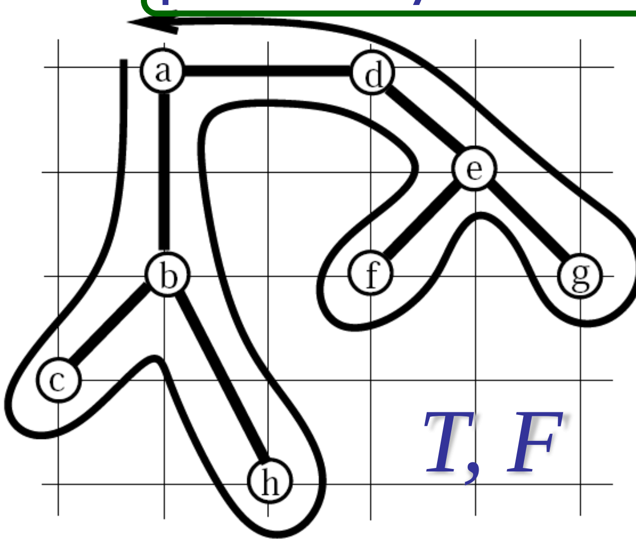
Every edge of T appears precisely **twice** in F:

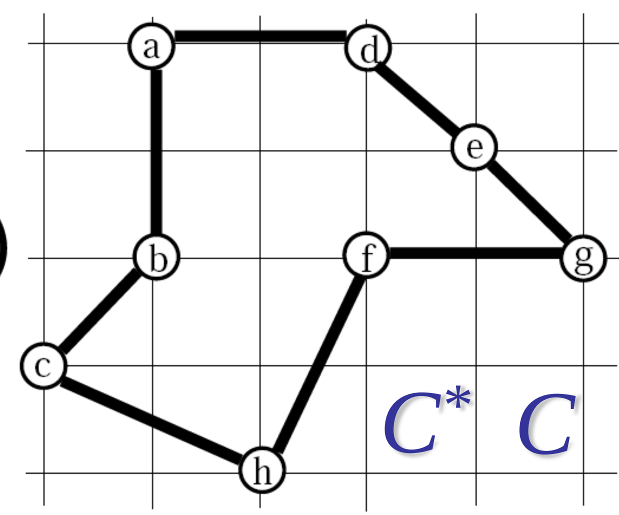
double tree algo

(ii) $L(F) = 2 \cdot L(T)$

(iii) $L(C) \leq L(F)$ because 3-inequal.

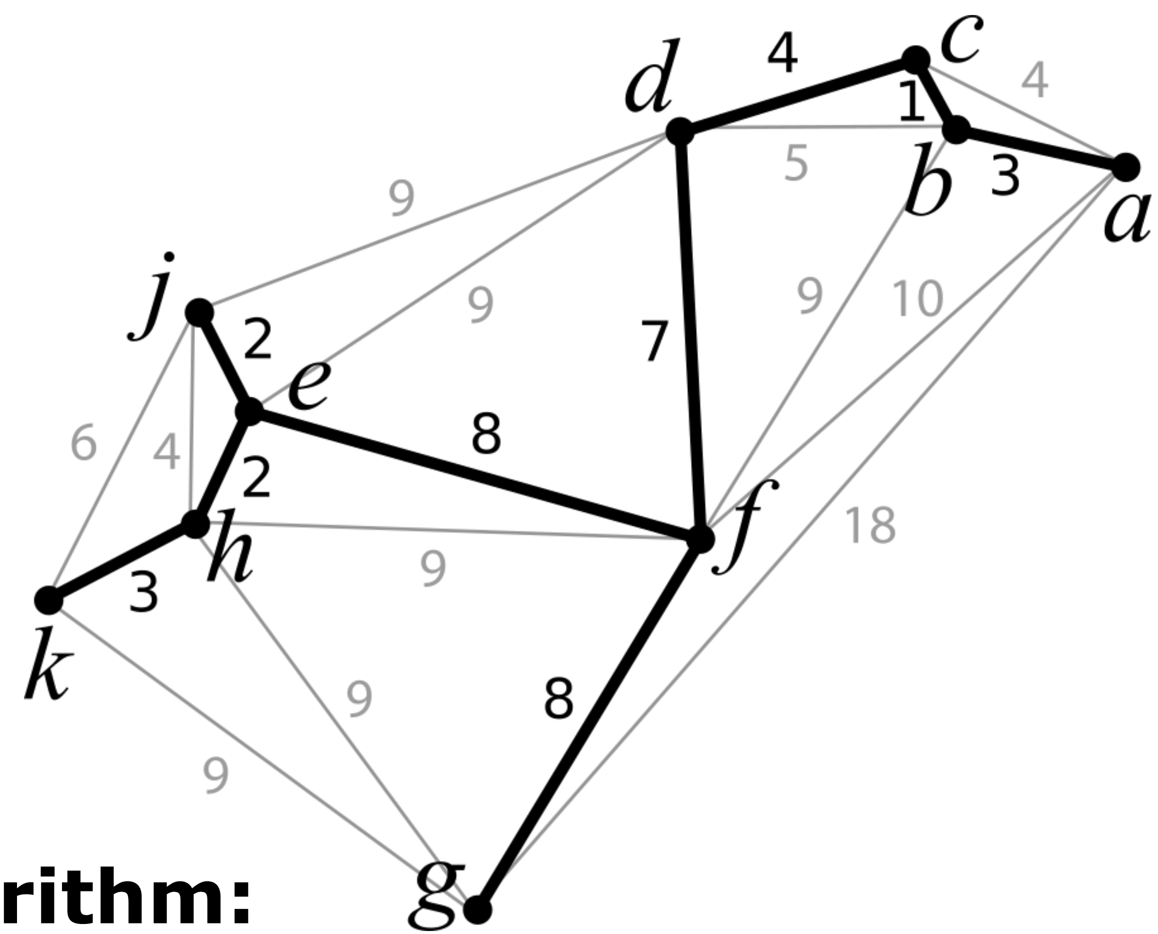
$\Rightarrow L(C) \leq L(F) = 2 \cdot L(T) \leq 2 \cdot L(C^*)$





Mini Quiz

Manually execute the *Double-Tree* Algorithm on this weighted graph:



***Double-Tree* Algorithm:**

Polytime approximating **minMTSP** up to factor 2:

1. Compute minimum spanning tree T of (G,w).
2. Traverse T <u>inorder</u> depth-first

Christofides Algorithm: 1.5

Input: $\underline{w}: V \times V \to \mathbb{N}$ edge weights <u>symmetric</u> and s.t. <u>triangle</u> inequality holds: $w(a,c) \leq w(a,b) + w(b,c)$.

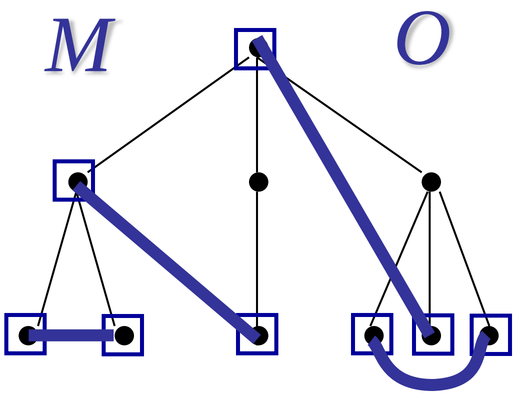
poly-time *Blossom* algorithm [Edmonds'65]

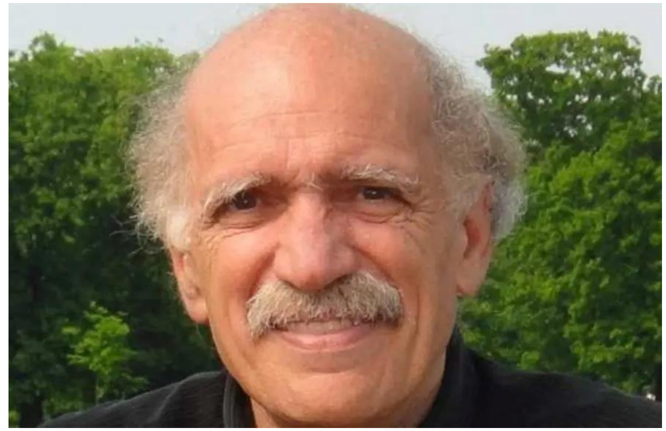
Polytime approximating **minMTSP** up to factor 1.5

1. Compute minimum spanning tree T of (V,w).
2. Let $O \subseteq V$ denote the vertices of *odd* degree in T
3. Find min.-weight perfect matching M of O wrt. w
4. Form *multi*-graph $T \cup M$
5. Calculate a *Eulerian* circuit $E \subseteq T \cup M$
6. Convert E into a *Hamiltonian* circuit C by skipping repeated vertices (*<u>shortcutting</u>*).

exists *iff* each vertex has *even* degree

Christofides *Illustrated*

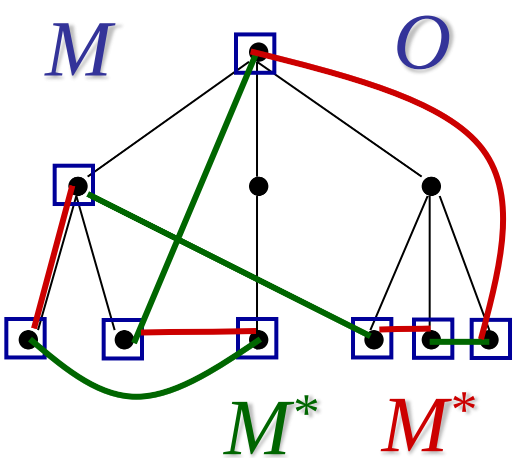




exists *why?*

1. Compute minimum spanning tree T of (V,w).
2. Let $O \subseteq V$ denote the vertices of *odd* degree in T
3. Find min.-weight perfect matching M of O wrt. w
4. Form *multi*-graph $T \cup M$: *all* vertices *even* degree
5. Calculate a *Eulerian* circuit E in $T \cup M$
6. Convert E into a *Hamiltonian* circuit C by skipping repeated vertices (*shortcutting*).

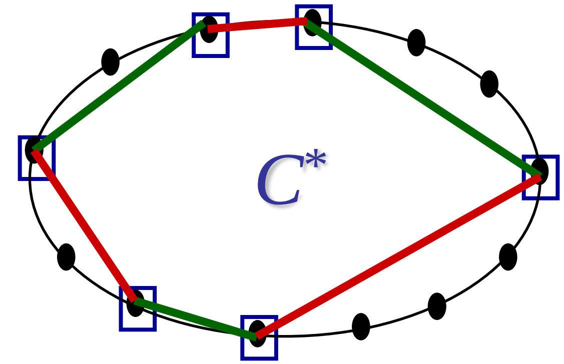
Christofides *Analysis*: 1.5



(i) $L(T) \le L(C^*)$

(ii) $L(M) \le \min\{L(M^*), L(M^*)\} \le L(C^*)/2$

(iii) $L(C) \le L(E) \le L(T) + L(M)$



$L(M^*) + L(M^*) \le L(C^*)$

1. Compute minimum spanning tree T of (V,w).
2. Let $O \subseteq V$ denote the vertices of *odd* degree in T
3. Find min.-weight perfect matching M of O wrt. w
4. Form *multi*-graph $T \cup M$
5. Calculate a *Eulerian* circuit E in $T \cup M$
6. Convert E into a *Hamiltonian* circuit C by skipping repeated vertices (*shortcutting*).

triangle inequality

Knapsack

Input: n packets, values $v_1,\ldots v_n \in \mathbb{N}$
and weights $w_1,\ldots w_n \in \mathbb{N}$
and *both* bounds V,W

Goal: find subset $S \subseteq \{1,\ldots,n\}$

- maximizing values $\sum_{p\in S} v_p$

subject to weight *upper* bound $\sum_{p\in S} w_p \leq W$

- minimizing weight $\sum_{p\in S} w_p$

subject to value *lower* bound $\sum_{p\in S} v_p \geq V$

Question: Does there *exist* a subset $S \subseteq \{1,\ldots,n\}$
such that values $\sum_{p\in S} v_p \geq V$ and weights $\sum_{p\in S} w_p \leq W$

Dynamic Programming: *Knapsack*

For $S \subseteq \{1,\ldots,n\}$ write $w(S)=\sum_{p\in S} w_p$ and $v(S)=\sum_{p\in S} v_p$

Goal: Given W, determine $V := \max\,\{v(S) : w(S)\leq W\}$

Consider $T(v,m) := \min\,\{\, w(S) : S\subseteq\{1,\ldots,m\},\ v(S)\geq v \,\}$

Note: i) $T(0,n) \leq T(1,n) \leq \ldots \leq T(V,n) \leq W < T(V+1,n)$

ii) $V = \max\,\{\, v : T(v,n) \leq W \,\}$

iii) $T(v,m) = 0$ for $v\leq 0$

iv) $T(v,0) = \infty$ for $v>0$

v) $T(v,m) = \min\,\{\, T(v,m-1)\,,\ w_m + T(v-v_m\,, m-1) \,\}$

w.l.o.g. $0<w_p\leq W$, $0<v_p\leq V$

$v \setminus m$	0	1	…	n
0	0	0	0	0
1	∞			
2	∞			
⋮	∞			
V	∞			

T

runtime poly$(n+V)$

Approximation Schemes

Input: n packets, values $v_1,\ldots v_n \in \mathbb{N}$
and weights $w_1,\ldots w_n \in \mathbb{N}$

and c) value bound V.

Question: Is there a subset
$S \subseteq \{1,\ldots,n\}$ s.t. values $\sum_{p\in S} v_p \geq V$
subject to weight bound $\sum_{p\in S} w_p \leq W$

Now: Find S' s.t. $\sum_{p\in S'} w_p \leq W$ and $\sum_{p\in S'} v_p \geq V\cdot(1-\varepsilon)$

Or: Find S'' s.t. $\sum_{p\in S''} w_p \leq W\cdot(1+\varepsilon)$ and $\sum_{p\in S''} v_p \geq V$

Algorithm: guaranteed approximation ratio $1\pm\varepsilon$

Discrete optimization → decision often $\mathcal{NP}$-hard
Try approximating maxim./minim. up to relative error

FPTAS for *Knapsack*

Scaling Lemma a) For $0 \leq \underline{v}' \leq \underline{v}$, $V(\underline{v}') \leq V(\underline{v})$

b) and for $\underline{v} \leq \underline{d} \leq (k,\ldots k)$: $V(\underline{v}-\underline{d}) \geq V(\underline{v}) - n \cdot k$

c) Also, $V(k \cdot \underline{v}) = k \cdot V(\underline{v})$

$v-k < \lfloor v/k \rfloor \cdot k \leq v$

Scaling Method: Fix $k \in \mathbb{N}$ and let $v_p' := \lfloor v_p / k \rfloor$

Compute $V' := k \cdot V(v_1', \ldots, v_n')$ in time $\text{poly}(n+V/k)$. So

$V' = V(\lfloor \underline{v}/k \rfloor \cdot k) \geq V(\underline{v}-k \cdot \underline{1}) \geq V-n \cdot k = V \cdot (1-n \cdot k/V) \overset{!}{\geq} V \cdot (1-\varepsilon)$

for $k :\approx \lfloor \varepsilon \cdot \sum_p v_p / n^2 \rfloor \leq \varepsilon \cdot V/n$

$0 < v_p \leq V \Rightarrow$
$V \leq \sum_p v_p \leq nV$

$V/k \leq O(n^2/\varepsilon + 1)$

Theorem: For every given $\varepsilon > 0$, can approximate Knapsack up to error $1-\varepsilon$ in time polynom. in $n+1/\varepsilon$

$V(v_1, \ldots v_n) := \max\{ \sum_{p \in S} v_p : S \subseteq \{1..n\}, \sum_{p \in S} w_p \leq W \}$

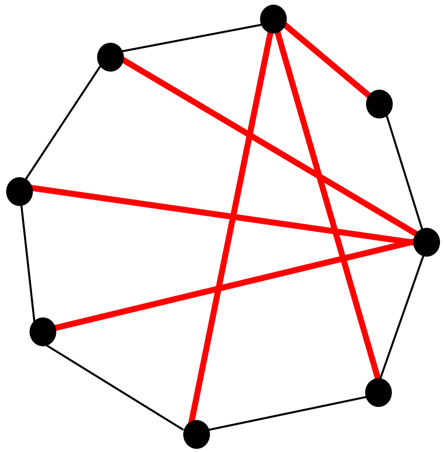
Limits of Approximation

Theorem: No polynom.-time algorithm can approximate the general TSP up to some constant unless $\mathcal{P}=\mathcal{NP}$.

Proof: Suppose $\mathcal{A}$ approximates TSP up to factor $c\in\mathbb{N}$.

Turn $\mathcal{A}$ into algorithm $\mathcal{B}$ for HC:

No triangle-inequality…



Algorithm $\mathcal{B}$, input graph $G=(V,E)$, $n:=|V|$.

Define $w(u,v) := 1$ for $\{u,v\}\in E$;
$w(u,v) := n\cdot c$ for $\{u,v\}\notin E$.

$\langle G\rangle\in\mathbf{HC} \Rightarrow$ w contains Hamiltonian cycle of weight n
$\Rightarrow$ algorithm $\mathcal{A}$ finds some of weight $\leq n\cdot c$

$\langle G\rangle\notin\mathbf{HC} \Rightarrow$ any Hamiltonian cycle has weight $> n\cdot c$

$\mathbf{HC} := \{ \langle G\rangle \mid$ graph G contains a Hamiltonian cycle $\}$
$\mathbf{TSP} := \{ \langle G,w,k\rangle \mid (G,w)$ contains a Hamiltonian cycle of weight $\leq k \}$

§8 Summary

- Motivation and Classification
- *metric* Travelling Salesperson
  - *Double-Tree*: approximation ratio 2
  - *Christofides*: approximation ratio 1.5
- Knapsack
  - Strongly polyn.-time / Dynamic Programming
  - *Fully* Polynomial-Time Approximation *Scheme*
- Limits of Approximability