Approximation Algorithms for Traveling Salesmen

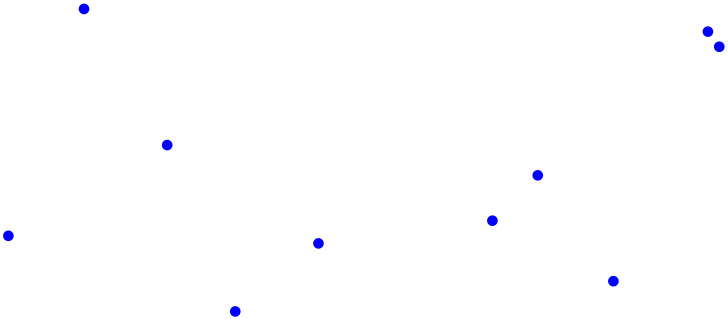
Jens Vygen

University of Bonn

Approximation Algorithms for Traveling Salesmen

Jens Vygen

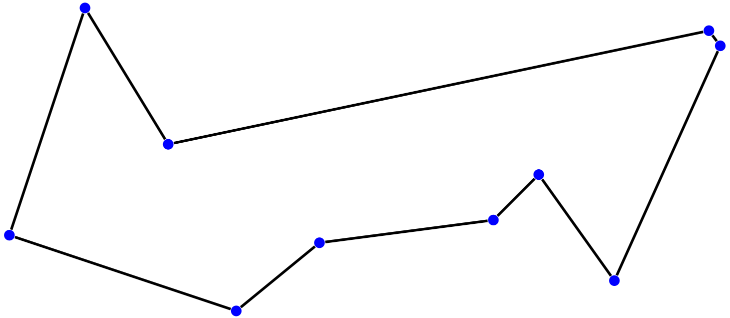
University of Bonn



Approximation Algorithms for Traveling Salesmen

Jens Vygen

University of Bonn



Variants of the TSP

- start = end?
- symmetric or asymmetric?
- triangle inequality?
- visit every city at least once or exactly once?

Variants of the TSP, Complexity

- start = end?
- symmetric or asymmetric?
- triangle inequality?
- visit every city at least once or exactly once?

All versions are *NP*-hard.

Variants of the TSP, Complexity

- start = end?
- symmetric or asymmetric?
- triangle inequality?
- visit every city at least once or exactly once?

All versions are *NP*-hard.

An f-approximation algorithm runs in polynomial time and always computes a tour that is at most f times longer than optimum. The best such f is called the approximation ratio.

Variants of the TSP, Complexity

- start = end?
- symmetric or asymmetric?
- triangle inequality?
- visit every city at least once or exactly once?

All versions are *NP*-hard.

An f-approximation algorithm runs in polynomial time and always computes a tour that is at most f times longer than optimum. The best such f is called the approximation ratio.

If no triangle inequality and must visit every city exactly once, no approximation algorithm exists unless $P = NP$.

Variants of the TSP, Complexity

- start = end?
- symmetric or asymmetric?
- triangle inequality?
- visit every city at least once or exactly once?

All versions are *NP*-hard.

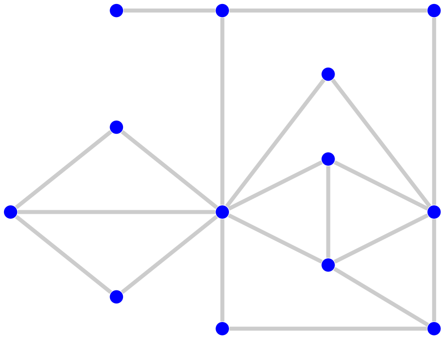
An f-approximation algorithm runs in polynomial time and always computes a tour that is at most f times longer than optimum. The best such f is called the approximation ratio.

If no triangle inequality and must visit every city exactly once, no approximation algorithm exists unless $P = NP$.

Whether we assume triangle inequality or allow visiting cities more than once is equivalent.

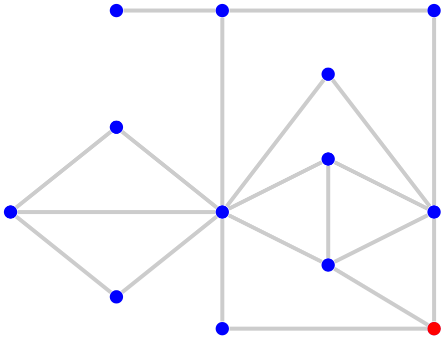
Example: Graph TSP

Given a connected graph G, find a minimum length closed edge progression in G that visits every vertex at least once.



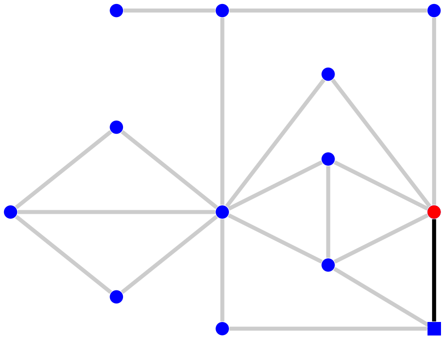
Example: Graph TSP

Given a connected graph G, find a minimum length closed edge progression in G that visits every vertex at least once.



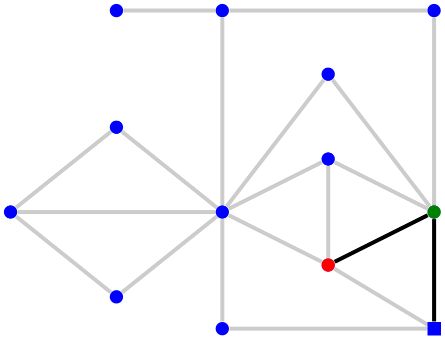
Example: Graph TSP

Given a connected graph G, find a minimum length closed edge progression in G that visits every vertex at least once.



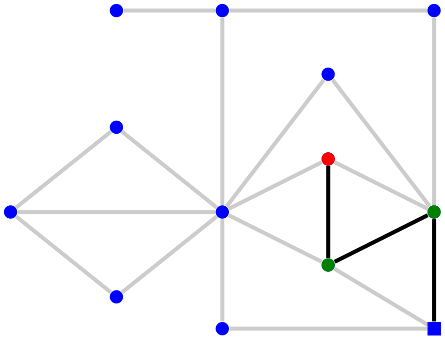
Example: Graph TSP

Given a connected graph G, find a minimum length closed edge progression in G that visits every vertex at least once.



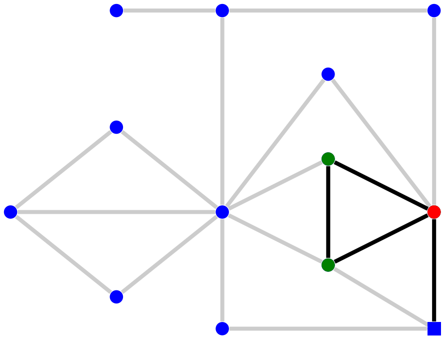
Example: Graph TSP

Given a connected graph G, find a minimum length closed edge progression in G that visits every vertex at least once.



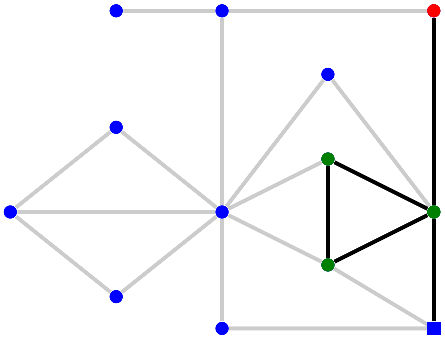
Example: Graph TSP

Given a connected graph G, find a minimum length closed edge progression in G that visits every vertex at least once.



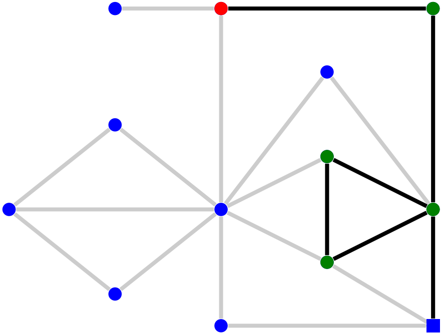
Example: Graph TSP

Given a connected graph G, find a minimum length closed edge progression in G that visits every vertex at least once.



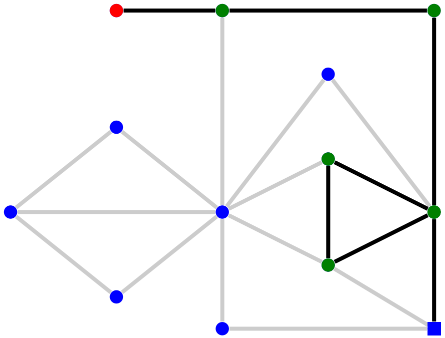
Example: Graph TSP

Given a connected graph G, find a minimum length closed edge progression in G that visits every vertex at least once.



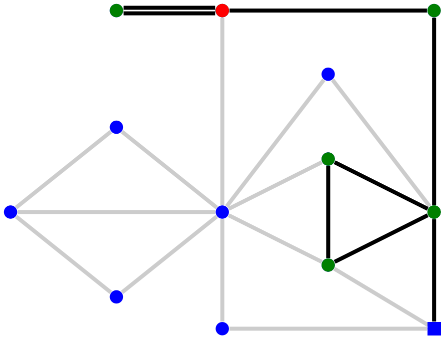
Example: Graph TSP

Given a connected graph G, find a minimum length closed edge progression in G that visits every vertex at least once.



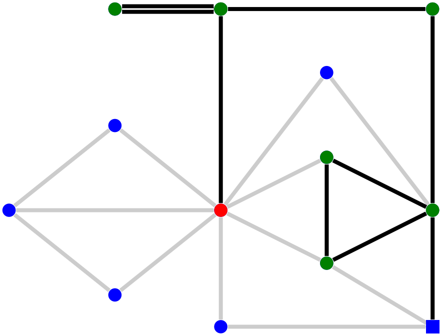
Example: Graph TSP

Given a connected graph G, find a minimum length closed edge progression in G that visits every vertex at least once.



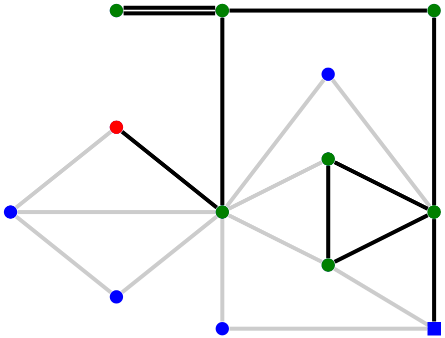
Example: Graph TSP

Given a connected graph G, find a minimum length closed edge progression in G that visits every vertex at least once.



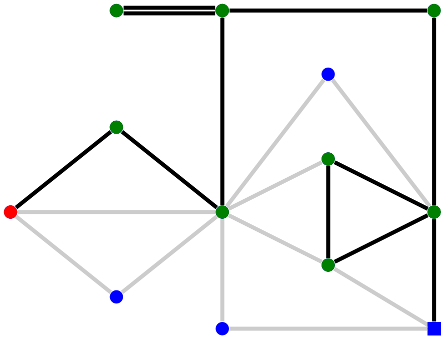
Example: Graph TSP

Given a connected graph G, find a minimum length closed edge progression in G that visits every vertex at least once.



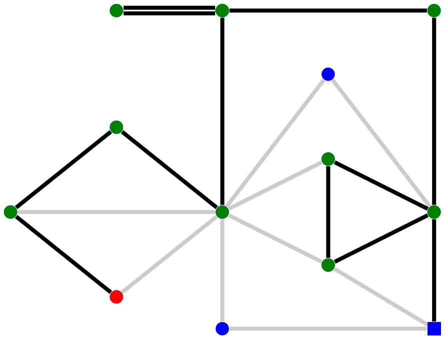
Example: Graph TSP

Given a connected graph G, find a minimum length closed edge progression in G that visits every vertex at least once.



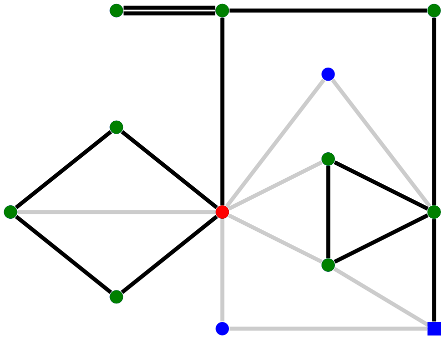
Example: Graph TSP

Given a connected graph G, find a minimum length closed edge progression in G that visits every vertex at least once.



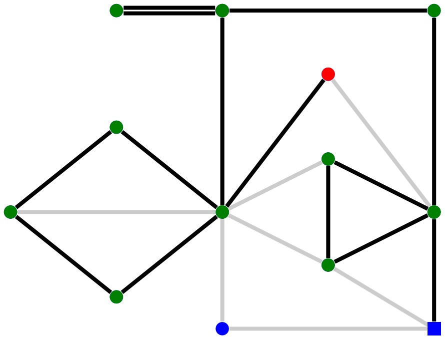
Example: Graph TSP

Given a connected graph G, find a minimum length closed edge progression in G that visits every vertex at least once.



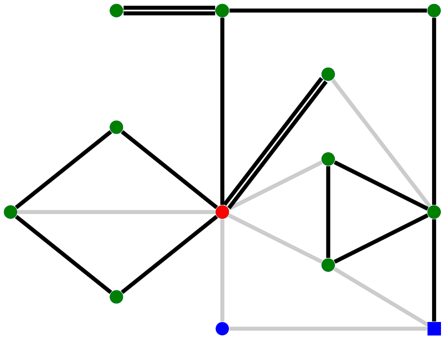
Example: Graph TSP

Given a connected graph G, find a minimum length closed edge progression in G that visits every vertex at least once.



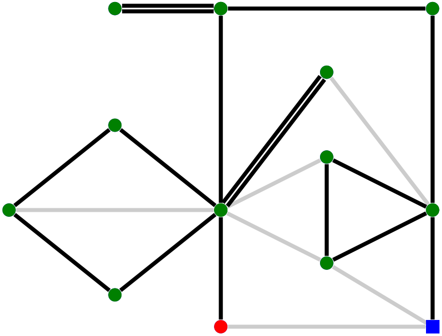
Example: Graph TSP

Given a connected graph G, find a minimum length closed edge progression in G that visits every vertex at least once.



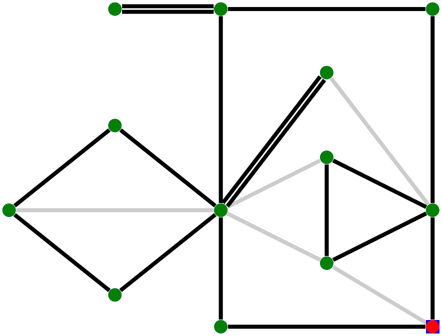
Example: Graph TSP

Given a connected graph G, find a minimum length closed edge progression in G that visits every vertex at least once.



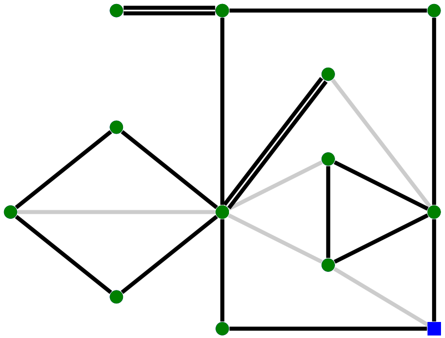
Example: Graph TSP

Given a connected graph G, find a minimum length closed edge progression in G that visits every vertex at least once.



Example: Graph TSP

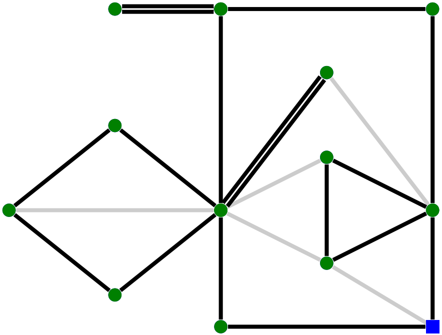
Given a connected graph G, find a minimum length closed edge progression in G that visits every vertex at least once.



Equivalently:

Example: Graph TSP

Given a connected graph G, find a minimum length closed edge progression in G that visits every vertex at least once.

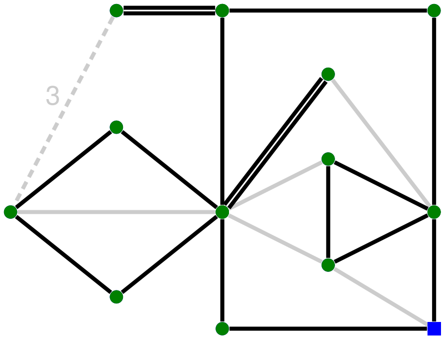


Equivalently:

- find a shortest Hamiltonian circuit in the metric closure of G

Example: Graph TSP

Given a connected graph G, find a minimum length closed edge progression in G that visits every vertex at least once.

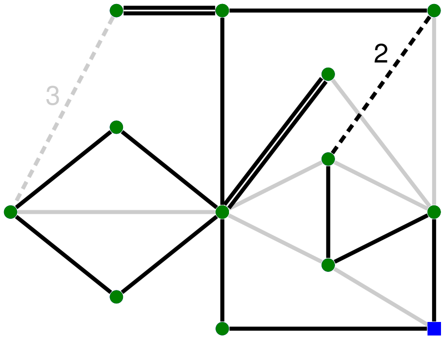


Equivalently:

- find a shortest Hamiltonian circuit in the metric closure of G

Example: Graph TSP

Given a connected graph G, find a minimum length closed edge progression in G that visits every vertex at least once.

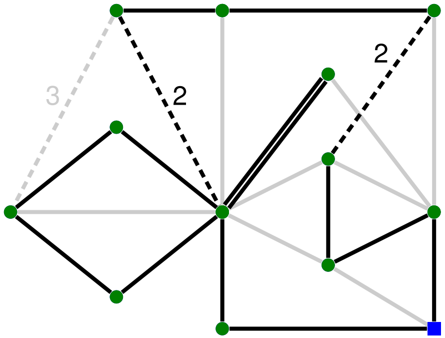


Equivalently:

- find a shortest Hamiltonian circuit in the metric closure of G

Example: Graph TSP

Given a connected graph G, find a minimum length closed edge progression in G that visits every vertex at least once.

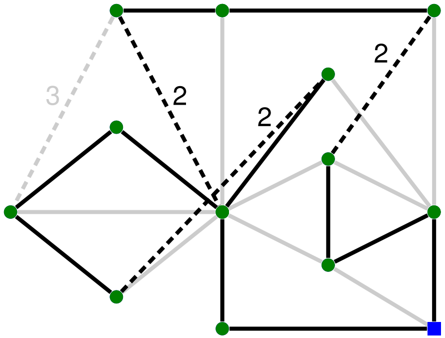


Equivalently:

- find a shortest Hamiltonian circuit in the metric closure of G

Example: Graph TSP

Given a connected graph G, find a minimum length closed edge progression in G that visits every vertex at least once.

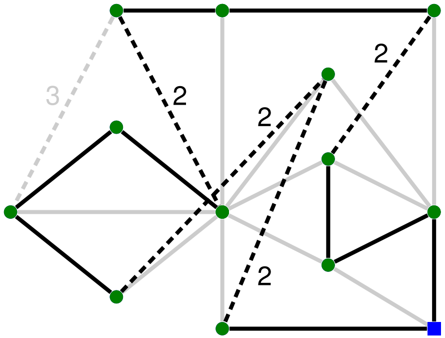


Equivalently:

- find a shortest Hamiltonian circuit in the metric closure of G

Example: Graph TSP

Given a connected graph G, find a minimum length closed edge progression in G that visits every vertex at least once.

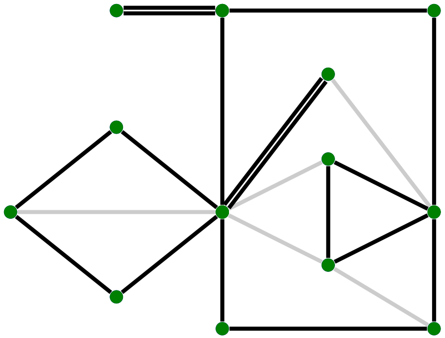


Equivalently:

- find a shortest Hamiltonian circuit in the metric closure of G

Example: Graph TSP

Given a connected graph G, find a minimum length closed edge progression in G that visits every vertex at least once.

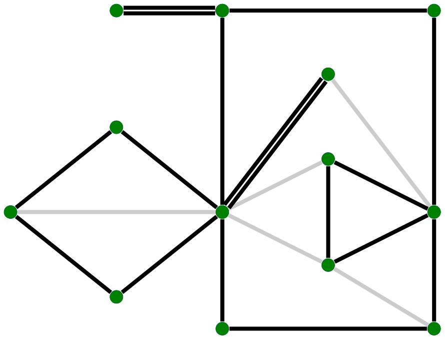


Equivalently:

- find a shortest Hamiltonian circuit in the metric closure of G
- find a smallest Eulerian spanning subgraph of $2G$

Example: Graph TSP

Given a connected graph G, find a minimum length closed edge progression in G that visits every vertex at least once.



Equivalently:

- find a shortest Hamiltonian circuit in the metric closure of G
- find a smallest $\underbrace{\text{Eulerian spanning subgraph of } 2G}_{\text{its edge set is also called tour}}$

Asymmetric TSP

Given a finite set V of cities and distances $c : V \times V \to \mathbb{R}_{\geq 0}$, find a tour (a list $v_0, \ldots, v_k$ containing each vertex at least once, with $v_0 = v_k$) of minimum total length $\sum_{i=1}^{k} c(v_{i-1}, v_i)$.

- $O(\log n)$-approximation algorithm, where $n = |V|$
  (Frieze, Galbiati, Maffioli [1982])
- $O(\log n / \log\log n)$-approximation algorithm
  (Asadpour, Goemans, Mądry, Oveis Gharan, Saberi [2010])
- no $\frac{75}{74}$-approximation algorithm exists unless $P = NP$
  (Karpinski, Lampis, Schmied [2013])
- integrality ratio between 2 and $\log^{O(1)} \log n$
  (Anari and Oveis Gharan [2015])

Essentially the same holds for the version where start $\neq$ end

Symmetric TSP: $c(v, w) = c(w, v)$ for all $v, w \in V$

- best known approximation ratio $\frac{3}{2}$
  (Christofides [1976])
- no $\frac{123}{122}$-approximation algorithm exists unless $P = NP$
  (Karpinski, Lampis, Schmied [2013])
- integrality ratio of subtour relaxation between $\frac{4}{3}$ and $\frac{3}{2}$
  (Wolsey [1980])

Symmetric TSP: $c(v,w) = c(w,v)$ for all $v, w \in V$

- best known approximation ratio $\frac{3}{2}$
  (Christofides [1976])
- no $\frac{123}{122}$-approximation algorithm exists unless $P = NP$
  (Karpinski, Lampis, Schmied [2013])
- integrality ratio of subtour relaxation between $\frac{4}{3}$ and $\frac{3}{2}$
  (Wolsey [1980])

Subtour relaxation (assuming triangle inequality):

$$\min \left\{ c(x) \;:\; x(\delta(v)) = 2\,(v \in V),\; x(\delta(U)) \geq 2\,(\emptyset \neq U \subset V),\; x \geq 0 \right\}$$

(Dantzig, Fulkerson, Johnson [1954], Held, Karp [1970],
Cornuéjols, Fonlupt, Naddef [1985], Cunningham; Monma, Munson, Pulleyblank [1990])

Symmetric TSP: $c(v,w) = c(w,v)$ for all $v, w \in V$

- best known approximation ratio $\frac{3}{2}$
  (Christofides [1976])
- no $\frac{123}{122}$-approximation algorithm exists unless $P = NP$
  (Karpinski, Lampis, Schmied [2013])
- integrality ratio of subtour relaxation between $\frac{4}{3}$ and $\frac{3}{2}$
  (Wolsey [1980])

$$\sum_{e=\{v,w\}\in\binom{V}{2}} c(v,w)x_e \qquad \sum_{v\in U,\, w\in V\setminus U} x_{\{v,w\}}$$

Subtour relaxation (assuming triangle inequality):

$$\min\left\{\, c(x) \;:\; x(\delta(v)) = 2\ (v \in V),\ \ x(\delta(U)) \geq 2\ (\emptyset \neq U \subset V),\ \ x \geq 0 \,\right\}$$

(Dantzig, Fulkerson, Johnson [1954], Held, Karp [1970],
Cornuéjols, Fonlupt, Naddef [1985], Cunningham; Monma, Munson, Pulleyblank [1990])

Symmetric TSP: $c(v,w) = c(w,v)$ for all $v, w \in V$

- best known approximation ratio $\frac{3}{2}$

  (Christofides [1976])
- no $\frac{123}{122}$-approximation algorithm exists unless $P = NP$

  (Karpinski, Lampis, Schmied [2013])
- integrality ratio of subtour relaxation between $\frac{4}{3}$ and $\frac{3}{2}$

  (Wolsey [1980])

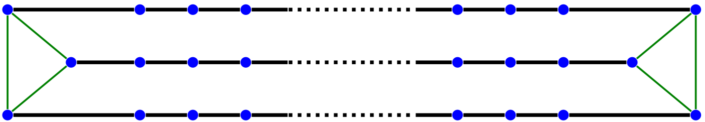
$\sum_{e=\{v,w\}\in\binom{V}{2}} c(v,w)x_e$ $\qquad$ $\sum_{v\in U,\, w\in V\setminus U} x_{\{v,w\}}$

Subtour relaxation (assuming triangle inequality):

$$\min\left\{\, c(x) \;:\; x(\delta(v)) = 2\ (v \in V),\ \ x(\delta(U)) \geq 2\ (\emptyset \neq U \subset V),\ \ x \geq 0 \,\right\}$$

(Dantzig, Fulkerson, Johnson [1954], Held, Karp [1970],

Cornuéjols, Fonlupt, Naddef [1985], Cunningham; Monma, Munson, Pulleyblank [1990])



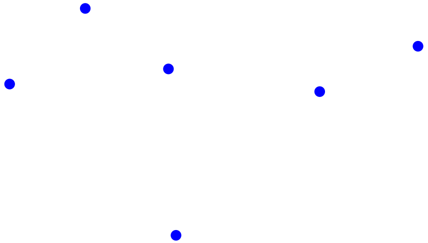
$c =$ graph distance

$x_e = \frac{1}{2}$

$x_e = 1$

Christofides' Algorithm

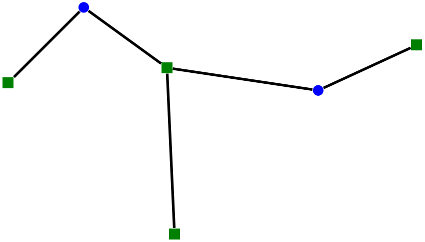
(Christofides [1976])



Christofides' Algorithm

(Christofides [1976])

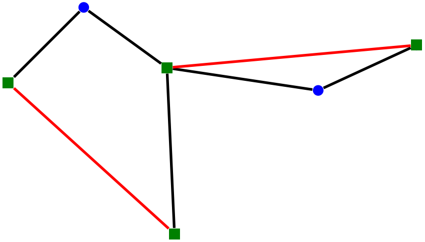
- Take a cheapest spanning tree (V, S)



Christofides' Algorithm

(Christofides [1976])

- Take a cheapest spanning tree (V, S)
- Do parity correction: add a cheapest T-join J, where T is the set of vertices with an odd degree in (V, S)

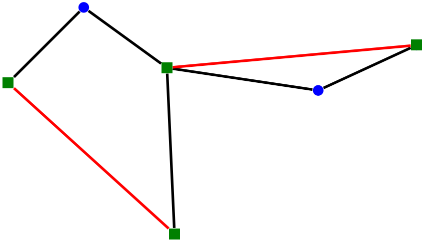


A T-join is a set J of edges such that T is the set of vertices with odd degree in (V, J). (Edmonds [1965])

Christofides' Algorithm

(Christofides [1976])

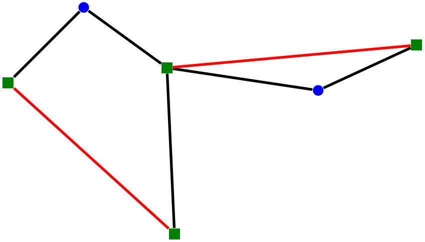
- Take a cheapest spanning tree (V, S)
- Do parity correction: add a cheapest T-join J, where T is the set of vertices with an odd degree in (V, S)
- Output $S + J$



A T-join is a set J of edges such that T is the set of vertices with odd degree in (V, J). (Edmonds [1965])

Christofides' Algorithm, Wolsey's Analysis (Wolsey [1980])

- Take a cheapest spanning tree (V, S)
- Do parity correction: add a cheapest T-join J, where T is the set of vertices with an odd degree in (V, S)
- Output $S + J$



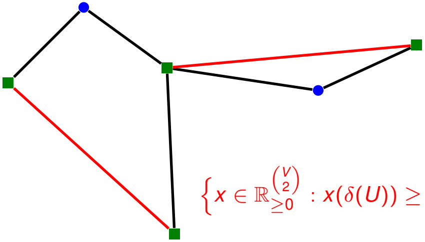
$\left\{x \in \mathbb{R}_{\geq 0}^{\binom{V}{2}} : x(\binom{U}{2}) \leq |U| - 1 \; (\emptyset \neq U \subset V),\; x(\binom{V}{2}) = |V| - 1\right\}$

Analysis:

- $\frac{n-1}{n} x^*$ is in the convex hull of spanning trees (Edmonds [1970])

Christofides' Algorithm, Wolsey's Analysis (Wolsey [1980])

- Take a cheapest spanning tree (V, S)
- Do parity correction: add a cheapest T-join J, where T is the set of vertices with an odd degree in (V, S)
- Output $S + J$



$$\left\{x \in \mathbb{R}_{\geq 0}^{\binom{V}{2}} : x(\delta(U)) \geq 1 \; (U \subset V, \; |T \cap U| \text{ odd})\right\}$$

$$\left\{x \in \mathbb{R}_{\geq 0}^{\binom{V}{2}} : x(\tbinom{U}{2}) \leq |U| - 1 \; (\emptyset \neq U \subset V), \; x(\tbinom{V}{2}) = |V| - 1\right\}$$

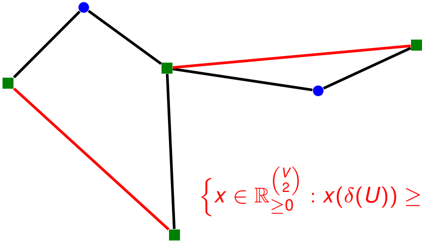
Analysis:

- $\frac{n-1}{n} x^*$ is in the convex hull of spanning trees (Edmonds [1970])
- $\frac{1}{2} x^*$ dominates a vector in the convex hull of T-joins

(Edmonds, Johnson [1973])

Christofides' Algorithm, Wolsey's Analysis (Wolsey [1980])

- Take a cheapest spanning tree (V, S)
- Do parity correction: add a cheapest T-join J, where T is the set of vertices with an odd degree in (V, S)
- Output $S + J$



$\left\{x \in \mathbb{R}_{\geq 0}^{\binom{V}{2}} : x(\delta(U)) \geq 1 \; (U \subset V, \; |T \cap U| \text{ odd})\right\}$

$\left\{x \in \mathbb{R}_{\geq 0}^{\binom{V}{2}} : x(\binom{U}{2}) \leq |U| - 1 \; (\emptyset \neq U \subset V), \; x(\binom{V}{2}) = |V| - 1\right\}$

Analysis:

- $\frac{n-1}{n} x^*$ is in the convex hull of spanning trees (Edmonds [1970])
- $\frac{1}{2} x^*$ dominates a vector in the convex hull of T-joins

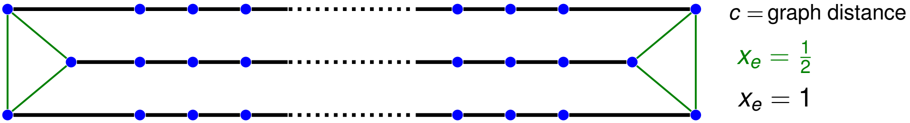
(Edmonds, Johnson [1973])

- $c(S + J) = c(S) + c(J) \leq \frac{n-1}{n} c(x^*) + \frac{1}{2} c(x^*) \leq \frac{3}{2} c(x^*)$.

Integrality ratio

Worst ratio of best integral solution (= optimum tour) and fractional solution (LP optimum)

- Wolsey's analysis shows an upper bound of $\frac{3}{2}$.
- These instances (of the Graph TSP) show a lower bound of $\frac{4}{3}$:



No better bounds are known!

Graph TSP

Given a graph $G = (V, E)$, let $c(v, w) =$ distance of v and w in G

Equivalently, look for a smallest Eulerian spanning subgraph of $2G$.

Improved approximation ratio for subcubic graphs:

- $\frac{4}{3}$ (Mömke, Svensson [2011])
  (before, for cubic graphs: (Boyd, Sitters, van der Ster, Stougie [2011]))
- $\frac{685}{684}$ impossible unless $P = NP$ (Karpinski, Schmied [2013])

Improved approximation ratios for general graphs:

- $1.5 - \epsilon$ (Oveis Gharan, Saberi, Singh [2011])
- 1.461 (Mömke, Svensson [2011])
- 1.445 (Mucha [2012])
- 1.4 (Sebő, V. [2014])

s-t-path TSP

Given a symmetric TSP instance and two cities s and t, find a shortest tour that begins in s and ends in t.

s-t-path TSP

Given a symmetric TSP instance and two cities s and t, find a shortest tour that begins in s and ends in t.

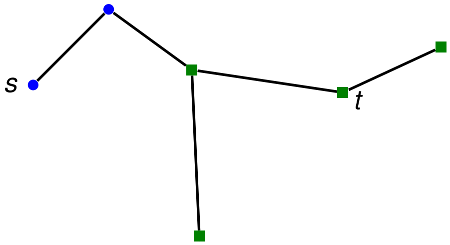
Equivalently, find a cheapest $\{s, t\}$-tour in $2G$, i.e., an $\{s, t\}$-join J such that (V, J) is connected

s-t-path TSP

Given a symmetric TSP instance and two cities s and t, find a shortest tour that begins in s and ends in t.

Equivalently, find a cheapest $\{s, t\}$-tour in $2G$, i.e., an $\{s, t\}$-join J such that (V, J) is connected

Can still do like Christofides, but now s and t must have odd degree (Hoogeveen [1991])



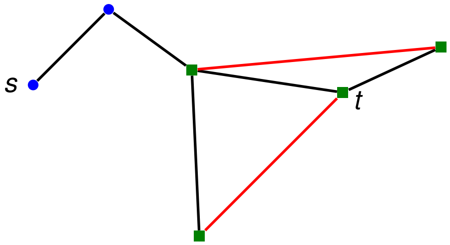
- take cheapest spanning tree S

s-t-path TSP

Given a symmetric TSP instance and two cities s and t, find a shortest tour that begins in s and ends in t.

Equivalently, find a cheapest $\{s, t\}$-tour in $2G$, i.e., an $\{s, t\}$-join J such that (V, J) is connected

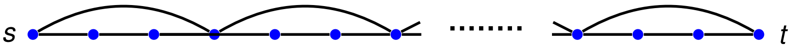
Can still do like Christofides, but now s and t must have odd degree (Hoogeveen [1991])



- take cheapest spanning tree S
- add cheapest T_S-join J

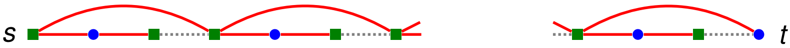
Lower bounds for the s-t-path TSP

Approximation ratio of Christofides/Hoogeveen is at least $\frac{5}{3}$:



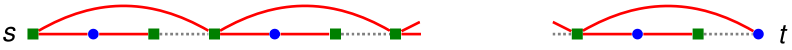
Lower bounds for the s-t-path TSP

Approximation ratio of Christofides/Hoogeveen is at least $\frac{5}{3}$:

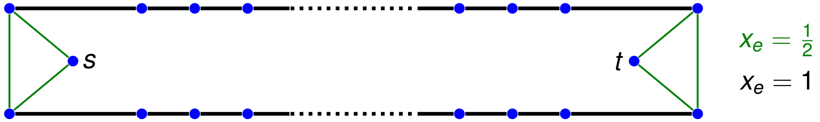


Lower bounds for the s-t-path TSP

Approximation ratio of Christofides/Hoogeveen is at least $\frac{5}{3}$:



Integrality ratio is at least $\frac{3}{2}$:



$$
\begin{array}{llcll}
\min & c(x) & & & \\
\text{subject to} & x(\delta(U)) & \geq & 2 & (\emptyset \neq U \subset V,\ |U \cap \{s,t\}| \text{ even}) \\
& x(\delta(U)) & \geq & 1 & (\emptyset \neq U \subset V,\ |U \cap \{s,t\}| \text{ odd}) \\
& x(\delta(v)) & = & 2 & (v \in V \setminus \{s,t\}) \\
& x(\delta(v)) & = & 1 & (v \in \{s,t\}) \\
& x & \geq & 0 &
\end{array}
$$

s-t-path TSP: approximation ratios

General symmetric weights:

- 1.667 (Hoogeveen [1991])
- 1.619 (An, Kleinberg, Shmoys [2012])
- 1.6 (Sebő [2013])
- 1.599 (V. [2015])
- 1.566 (Gottschalk, V. [2016]) $\longleftarrow$
- 1.53 (Sebő, van Zuylen [2016]

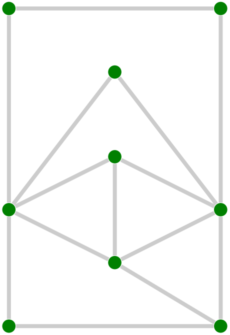
In graphs:

- 1.586 (Mömke, Svensson [2011])
- 1.584 (Mucha [2012])
- 1.578 (An, Kleinberg, Shmoys [2012])
- 1.5 (Sebő, V. [2014]) $\longleftarrow$

Ear-decompositions

Write $G = P_0 + P_1 + \cdots + P_k$, where P_0 is a single vertex, and each P_i ($i = 1, \ldots, k$) is either

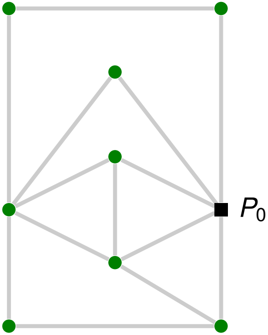
- a circuit sharing exactly one vertex with $P_0 + \cdots + P_{i-1}$, or
- a path sharing exactly its endpoints with $P_0 + \cdots + P_{i-1}$.



Ear-decompositions

Write $G = P_0 + P_1 + \cdots + P_k$, where P_0 is a single vertex, and each P_i ($i = 1, \ldots, k$) is either

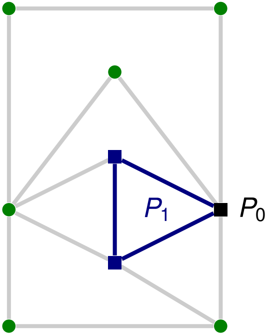
- a circuit sharing exactly one vertex with $P_0 + \cdots + P_{i-1}$, or
- a path sharing exactly its endpoints with $P_0 + \cdots + P_{i-1}$.



Ear-decompositions

Write $G = P_0 + P_1 + \cdots + P_k$, where P_0 is a single vertex, and each P_i ($i = 1, \ldots, k$) is either

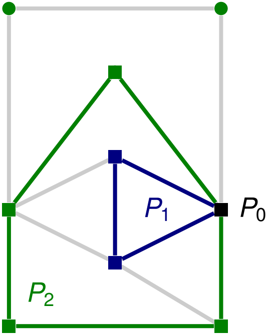
- a circuit sharing exactly one vertex with $P_0 + \cdots + P_{i-1}$, or
- a path sharing exactly its endpoints with $P_0 + \cdots + P_{i-1}$.



Ear-decompositions

Write $G = P_0 + P_1 + \cdots + P_k$, where P_0 is a single vertex, and each P_i ($i = 1, \ldots, k$) is either

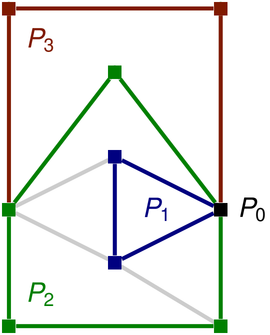
- a circuit sharing exactly one vertex with $P_0 + \cdots + P_{i-1}$, or
- a path sharing exactly its endpoints with $P_0 + \cdots + P_{i-1}$.



Ear-decompositions

Write $G = P_0 + P_1 + \cdots + P_k$, where P_0 is a single vertex, and each P_i ($i = 1, \ldots, k$) is either

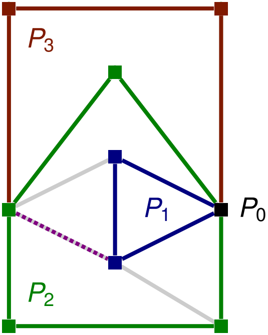
- a circuit sharing exactly one vertex with $P_0 + \cdots + P_{i-1}$, or
- a path sharing exactly its endpoints with $P_0 + \cdots + P_{i-1}$.



Ear-decompositions

Write $G = P_0 + P_1 + \cdots + P_k$, where P_0 is a single vertex, and each P_i ($i = 1, \ldots, k$) is either

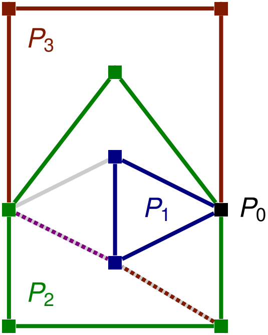
- a circuit sharing exactly one vertex with $P_0 + \cdots + P_{i-1}$, or
- a path sharing exactly its endpoints with $P_0 + \cdots + P_{i-1}$.



Ear-decompositions

Write $G = P_0 + P_1 + \cdots + P_k$, where P_0 is a single vertex, and each P_i ($i = 1, \ldots, k$) is either

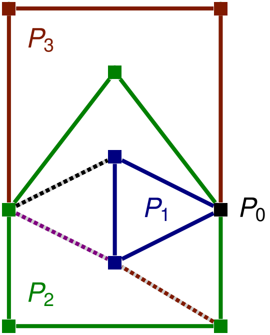
- a circuit sharing exactly one vertex with $P_0 + \cdots + P_{i-1}$, or
- a path sharing exactly its endpoints with $P_0 + \cdots + P_{i-1}$.



Ear-decompositions

Write $G = P_0 + P_1 + \cdots + P_k$, where P_0 is a single vertex, and each P_i ($i = 1, \ldots, k$) is either

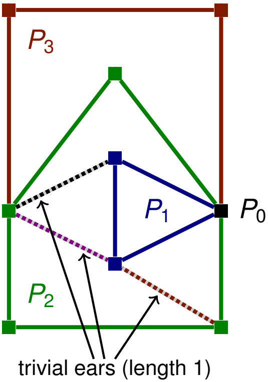
- a circuit sharing exactly one vertex with $P_0 + \cdots + P_{i-1}$, or
- a path sharing exactly its endpoints with $P_0 + \cdots + P_{i-1}$.



Ear-decompositions

Write $G = P_0 + P_1 + \cdots + P_k$, where P_0 is a single vertex, and each P_i ($i = 1, \ldots, k$) is either

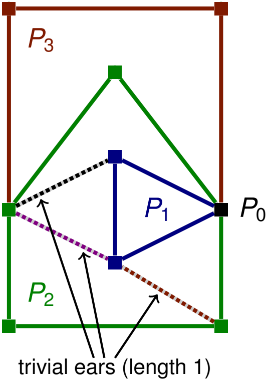
- a circuit sharing exactly one vertex with $P_0 + \cdots + P_{i-1}$, or
- a path sharing exactly its endpoints with $P_0 + \cdots + P_{i-1}$.



Ear-decompositions

Write $G = P_0 + P_1 + \cdots + P_k$, where P_0 is a single vertex, and each P_i ($i = 1, \ldots, k$) is either

- a circuit sharing exactly one vertex with $P_0 + \cdots + P_{i-1}$, or
- a path sharing exactly its endpoints with $P_0 + \cdots + P_{i-1}$.

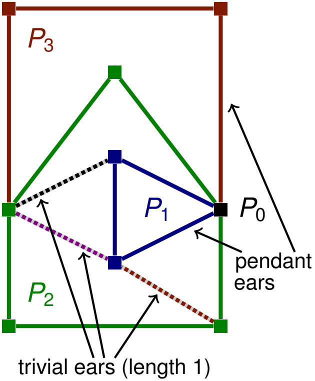


- A graph is 2-edge-connected iff it has an ear-decomposition.

Ear-decompositions

Write $G = P_0 + P_1 + \cdots + P_k$, where P_0 is a single vertex, and each P_i ($i = 1, \ldots, k$) is either

- a circuit sharing exactly one vertex with $P_0 + \cdots + P_{i-1}$, or
- a path sharing exactly its endpoints with $P_0 + \cdots + P_{i-1}$.

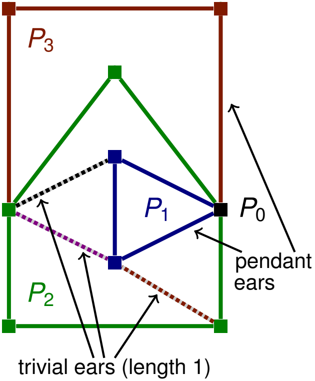


- A graph is 2-edge-connected iff it has an ear-decomposition.
- A nontrivial ear is called pendant if none of its internal vertices is endpoint of another nontrivial ear.

Ear-decompositions

Write $G = P_0 + P_1 + \cdots + P_k$, where P_0 is a single vertex, and each P_i ($i = 1, \ldots, k$) is either

- a circuit sharing exactly one vertex with $P_0 + \cdots + P_{i-1}$, or
- a path sharing exactly its endpoints with $P_0 + \cdots + P_{i-1}$.

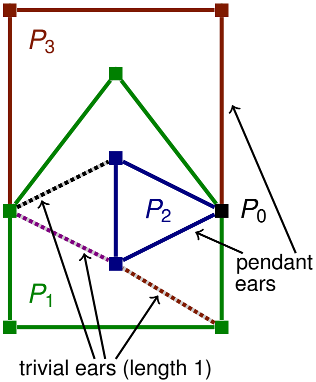


- A graph is 2-edge-connected iff it has an ear-decomposition.
- A nontrivial ear is called pendant if none of its internal vertices is endpoint of another nontrivial ear.
- W.l.o.g., pendant ears come last, followed only by trivial ears.

Ear-decompositions

Write $G = P_0 + P_1 + \cdots + P_k$, where P_0 is a single vertex, and each P_i ($i = 1, \ldots, k$) is either

- a circuit sharing exactly one vertex with $P_0 + \cdots + P_{i-1}$, or
- a path sharing exactly its endpoints with $P_0 + \cdots + P_{i-1}$.



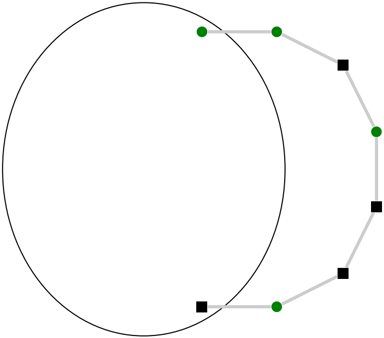
- A graph is 2-edge-connected iff it has an ear-decomposition.
- A nontrivial ear is called pendant if none of its internal vertices is endpoint of another nontrivial ear.
- W.l.o.g., pendant ears come last, followed only by trivial ears.

Ear induction for parity correction

For every T,

$$\min\{|J| : J \text{ is a } T\text{-join}\} \leq \frac{1}{2}(n - 1 + k_{\text{even}}),$$

where k_{even} is the number of even ears.



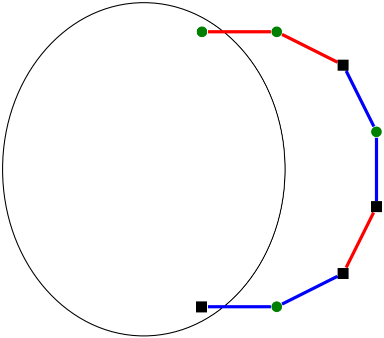
Proof:

Ear induction for parity correction

For every T,

$$\min\{|J| : J \text{ is a } T\text{-join}\} \;\leq\; \tfrac{1}{2}(n-1+k_{\text{even}}),$$

where k_{even} is the number of even ears.



Proof:

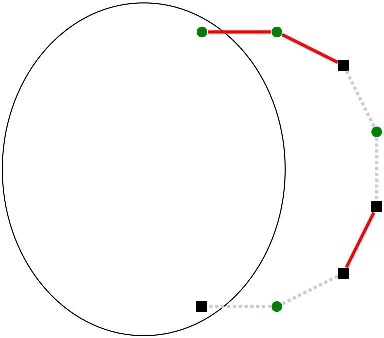
- Split pendant ear P at the vertices that have the wrong parity so far into red and blue part

Ear induction for parity correction

For every T,

$$\min\{|J| : J \text{ is a } T\text{-join}\} \;\le\; \tfrac{1}{2}(n - 1 + k_{\text{even}}),$$

where k_{even} is the number of even ears.



$\text{in}(P) :=$ number of inner vertices of P

$$k_{\text{even}}(P) := \begin{cases} 1 & \textit{if } \text{P } \textit{even} \\ 0 & \textit{if } \text{P } \textit{odd} \end{cases}$$

Proof:

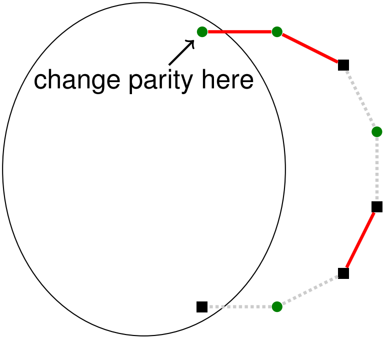
- Split pendant ear P at the vertices that have the wrong parity so far into red and blue part
- Take the smaller part, with $\le \frac{1}{2}(\text{in}(P) + k_{\text{even}}(P))$ edges

Ear induction for parity correction

For every T,

$$\min\{|J| : J \text{ is a } T\text{-join}\} \ \le\ \tfrac{1}{2}(n - 1 + k_{\text{even}}),$$

where k_{even} is the number of even ears.



$\text{in}(P) :=$ number of inner vertices of P

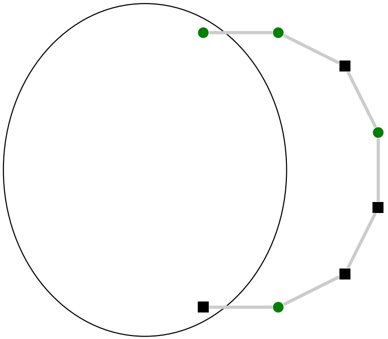
$$k_{\text{even}}(P) := \begin{cases} 1 & \textit{if P even} \\ 0 & \textit{if P odd} \end{cases}$$

Proof:

- Split pendant ear P at the vertices that have the wrong parity so far into red and blue part
- Take the smaller part, with $\le \frac{1}{2}(\text{in}(P) + k_{\text{even}}(P))$ edges
- Change parity of an endpoint of P if necessary; delete P; iterate $\square$

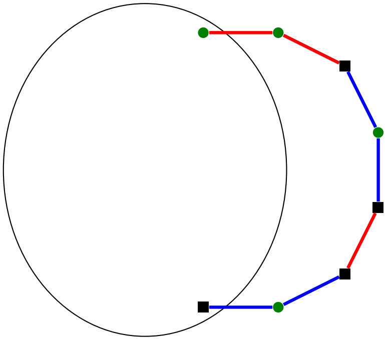
Ear induction for tours

Compute a tour with at most $\frac{3}{2}(n-1)+\frac{1}{2}\left(k_2-k_{\geq 4}\right)$ edges:



Ear induction for tours

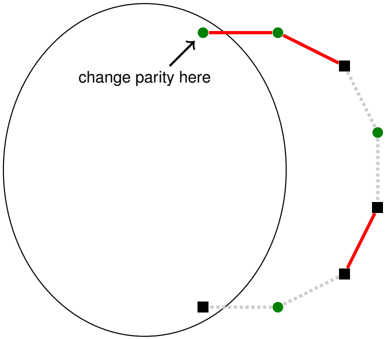
Compute a tour with at most $\frac{3}{2}(n-1)+\frac{1}{2}\left(k_2-k_{\geq 4}\right)$ edges:



- Split ear at the vertices that have wrong parity so far.

Ear induction for tours

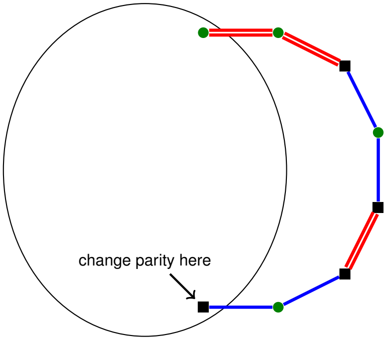
Compute a tour with at most $\frac{3}{2}(n-1) + \frac{1}{2}(k_2 - k_{\geq 4})$ edges:



- Split ear at the vertices that have wrong parity so far.
- Take smaller part for obtaining a T-join.

Ear induction for tours

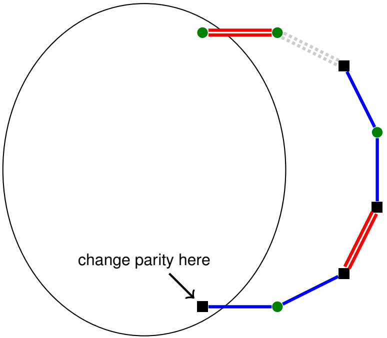
Compute a tour with at most $\frac{3}{2}(n-1)+\frac{1}{2}\left(k_2-k_{\geq 4}\right)$ edges:



- Split ear at the vertices that have wrong parity so far.
- Take smaller part for obtaining a T-join.
- Double smaller part for obtaining a T-tour.

Ear induction for tours

Compute a tour with at most $\frac{3}{2}(n-1) + \frac{1}{2}\left(k_2 - k_{\geq 4}\right)$ edges:



- Split ear at the vertices that have wrong parity so far.
- Take smaller part for obtaining a T-join.
- Double smaller part for obtaining a T-tour.
- May delete one pair of parallel edges (if there is one).

Need at most $\frac{3}{2}|\mathrm{in}(P)| - 1 + \frac{1}{2}k_{\mathrm{even}}(P)$ edges, or $\mathrm{in}(P) + 1$.
This is at most $\frac{3}{2}|\mathrm{in}(P)| + \frac{1}{2}\left(k_2(P) - k_{\geq 4}(P)\right)$. $\square$

Sketch of the first $\frac{3}{2}$-approximation algorithm for s-t-path TSP in graphs (Sebő, V. [2014])

- Compute an ear-decomposition in which the 2-ears are pendant and form a forest (using matroid intersection).
- If this is impossible, use Rado's theorem to get a stronger lower bound (details omitted).

Sketch of the first $\frac{3}{2}$-approximation algorithm for s-t-path TSP in graphs

(Sebő, V. [2014])

- Compute an ear-decomposition in which the 2-ears are pendant and form a forest (using matroid intersection).
- If this is impossible, use Rado's theorem to get a stronger lower bound (details omitted).
- Now two constructions:
- (1) Ear induction yields a tour with at most $\frac{3}{2}(n-1) + \frac{1}{2}\big(k_2 - k_{\geq 4}\big)$ edges. Good if $k_2 \leq k_{\geq 4}$.

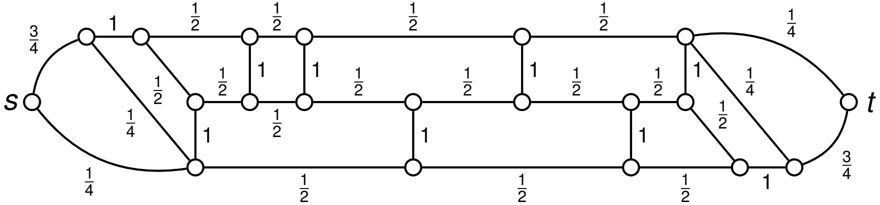
Sketch of the first $\frac{3}{2}$-approximation algorithm for s-t-path TSP in graphs

(Sebő, V. [2014])

- Compute an ear-decomposition in which the 2-ears are pendant and form a forest (using matroid intersection).
- If this is impossible, use Rado's theorem to get a stronger lower bound (details omitted).
- Now two constructions:
- (1) Ear induction yields a tour with at most $\frac{3}{2}(n-1) + \frac{1}{2}\left(k_2 - k_{\geq 4}\right)$ edges. Good if $k_2 \leq k_{\geq 4}$.
- (2) Take the 2-ears (but only one edge if s or t is the middle vertex), add edges for connectivity, and do parity correction. Yields a tour with at most $n - 1 + \frac{1}{2}\left(n - k_2 - 1 + k_{\geq 4}\right)$ edges. Good if $k_2 \geq k_{\geq 4}$.
- The better of the two tours has at most $\frac{3}{2}(n-1)$ edges.

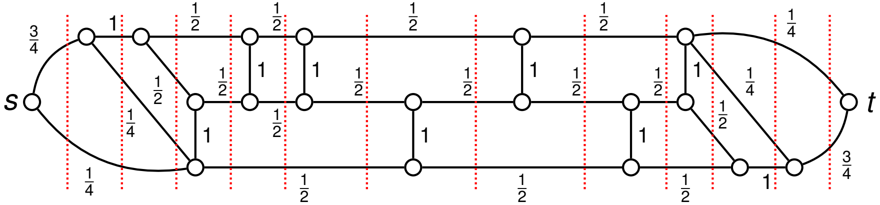
LP relaxation for s-t-path TSP

$$
\begin{array}{llcll}
\text{min} & c(x) & & & \\
\text{subject to} & x(\delta(U)) & \geq & 2 & (\emptyset \neq U \subset V,\ |U \cap \{s,t\}| \text{ even}) \\
& x(\delta(U)) & \geq & 1 & (\emptyset \neq U \subset V,\ |U \cap \{s,t\}| \text{ odd}) \\
& x(\delta(v)) & = & 2 & (v \in V \setminus \{s,t\}) \\
& x(\delta(v)) & = & 1 & (v \in \{s,t\}) \\
& x & \geq & 0 &
\end{array}
$$



LP relaxation for s-t-path TSP

$$
\begin{array}{llcll}
\min & c(x) & & & \\
\text{subject to} & x(\delta(U)) & \geq & 2 & (\emptyset \neq U \subset V,\ |U \cap \{s,t\}| \text{ even}) \\
& x(\delta(U)) & \geq & 1 & (\emptyset \neq U \subset V,\ |U \cap \{s,t\}| \text{ odd}) \\
& x(\delta(v)) & = & 2 & (v \in V \setminus \{s,t\}) \\
& x(\delta(v)) & = & 1 & (v \in \{s,t\}) \\
& x & \geq & 0 &
\end{array}
$$

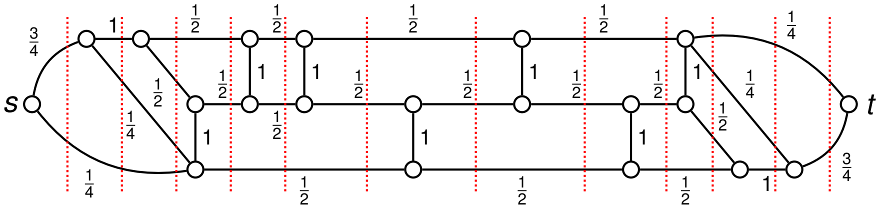


Cuts $C = \delta(U)$ with $x^*(C) < 2$ are called *narrow*. They form a chain.

Second $\frac{3}{2}$-approximation algorithm for s-t-path TSP in graphs

(Gao [2013])

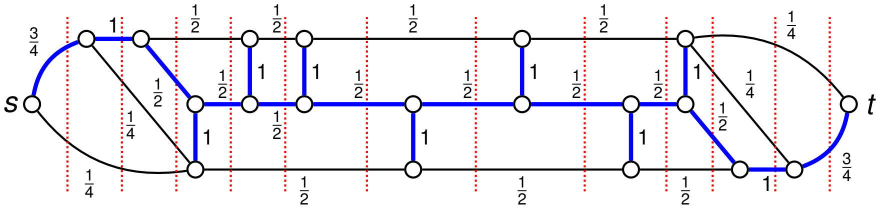
- Solve the LP. Let x^* be an optimum solution.
- Gao's Theorem: There is a spanning tree (V, S) in the support that contains only one edge in every narrow cut. ("Gao tree")



Second $\frac{3}{2}$-approximation algorithm for s-t-path TSP in graphs

(Gao [2013])

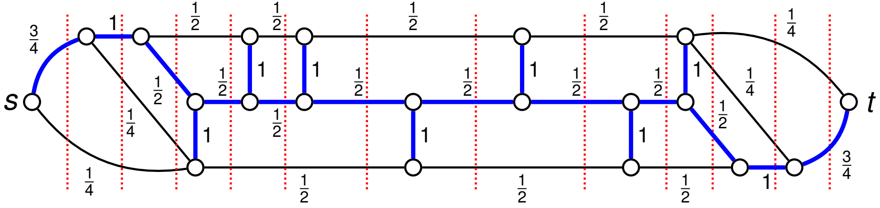
- Solve the LP. Let x^* be an optimum solution.
- Gao's Theorem: There is a spanning tree (V, S) in the support that contains only one edge in every narrow cut. ("Gao tree")



Second $\frac{3}{2}$-approximation algorithm for s-t-path TSP in graphs

(Gao [2013])

- Solve the LP. Let x^* be an optimum solution.
- Gao's Theorem: There is a spanning tree (V, S) in the support that contains only one edge in every narrow cut. ("Gao tree")
- Do parity correction. This costs at most $\frac{1}{2}c(x^*)$, because $\frac{1}{2}x^*$ dominates a vector in the convex hull of T_S-joins, where T_S is the set of vertices whose degree in S has the wrong parity
- Total number of edges at most $n - 1 + \frac{1}{2}c(x^*)$.



Now: general metrics

Most new algorithms (for all TSP variants) for general metrics

- first solve the natural LP relaxation,
- write the solution x^* as convex combination (=distribution) of spanning trees,
- sample a spanning tree from this distribution,
- and do parity correction.

Now: general metrics

Most new algorithms (for all TSP variants) for general metrics

- first solve the natural LP relaxation,
- write the solution x^* as convex combination (=distribution) of spanning trees,
- sample a spanning tree from this distribution,
- and do parity correction.

A deterministic variant tries all spanning trees with positive coefficient (less than n^2).

Now: general metrics

Most new algorithms (for all TSP variants) for general metrics

- first solve the natural LP relaxation,
- write the solution x^* as convex combination (=distribution) of spanning trees,
- sample a spanning tree from this distribution,
- and do parity correction.

A deterministic variant tries all spanning trees with positive coefficient (less than n^2).

Different distributions were used:

- max entropy distribution (Asadpour et al. [2010], Oveis Gharan et al. [2011])
- arbitrary distribution (An, Kleinberg, Shmoys [2012], Sebő [2013])
- distribution improved by local reassembling (V. [2015])
- Gao tree distribution (Gottschalk, V. [2016])

Best-of-Many-Christofides (An, Kleinberg, Shmoys [2012])

- Solve the LP, let x^* be an optimum solution
- Decompose x^* into spanning trees: write

$$x^* = \sum_{S \in \mathcal{S}} p_S \chi^S$$

  where $p_S \geq 0$ ($S \in \mathcal{S}$) and $\sum_{S \in \mathcal{S}} p_S = 1$
- Do parity correction for each $S \in \mathcal{S}$ with $p_S > 0$: add a minimum cost T_S-join
- Take the best of these tours

$\mathcal{S}$ is the set of edge sets of spanning trees

T_S is the set of vertices whose degree in S has the wrong parity (even for s or t, odd for other vertices)

Basic analysis

(An, Kleinberg, Shmoys [2012])

The result has cost

$$
\begin{aligned}
& \min_{S \in \mathcal{S}:\, p_S > 0} \big(c(S) + \min\{c(J) : J \text{ is a } T_S\text{-join}\} \big) \\
\leq \;& \sum_{S \in \mathcal{S}} p_S \big(c(S) + \min\{c(J) : J \text{ is a } T_S\text{-join}\} \big) \\
= \;& c(x^*) + \sum_{S \in \mathcal{S}} p_S \min\{c(J) : J \text{ is a } T_S\text{-join}\}
\end{aligned}
$$

Basic analysis

(An, Kleinberg, Shmoys [2012])

The result has cost

$$
\begin{aligned}
& \min_{S\in\mathcal{S}:\, p_S>0} \big(c(S) + \min\{c(J) : J \text{ is a } T_S\text{-join}\}\big) \\
\leq \quad & \sum_{S\in\mathcal{S}} p_S \big(c(S) + \min\{c(J) : J \text{ is a } T_S\text{-join}\}\big) \\
= \quad & c(x^*) + \sum_{S\in\mathcal{S}} p_S \min\{c(J) : J \text{ is a } T_S\text{-join}\} \\
\leq \quad & c(x^*) + \sum_{S\in\mathcal{S}} p_S c(y^S)
\end{aligned}
$$

for any set of correction vectors y^S ($S \in \mathcal{S}$) such that y^S is in the T_S-join polyhedron

$$
\Big\{ y \in \mathbb{R}^E_{\geq 0} : y(C) \geq 1 \ \forall\ T_S\text{-cuts } C \Big\}
$$

(Edmonds, Johnson [1973])

Basic analysis

(An, Kleinberg, Shmoys [2012])

The result has cost

$$\begin{aligned}
&\min_{S\in\mathcal{S}:\, p_S>0}\big(c(S)+\min\{c(J) : J \text{ is a } T_S\text{-join}\}\big)\\
\le\;& \sum_{S\in\mathcal{S}} p_S\big(c(S)+\min\{c(J) : J \text{ is a } T_S\text{-join}\}\big)\\
=\;& c(x^*)+\sum_{S\in\mathcal{S}} p_S \min\{c(J) : J \text{ is a } T_S\text{-join}\}\\
\le\;& c(x^*)+\sum_{S\in\mathcal{S}} p_S c(y^S)
\end{aligned}$$

for any set of correction vectors y^S ($S\in\mathcal{S}$) such that y^S is in the T_S-join polyhedron

$$\left\{y\in\mathbb{R}^E_{\ge 0} : y(C)\ge 1\ \forall\ T_S\text{-cuts } C\right\}$$

(Edmonds, Johnson [1973])

Example: x^* is a correction vector for every S, and $\frac{x^*}{2}$ almost

Basic analysis

(An, Kleinberg, Shmoys [2012])

The result has cost

$$
\begin{aligned}
& \min_{S \in \mathcal{S}:\, p_S > 0} \big(c(S) + \min\{c(J) : J \text{ is a } T_S\text{-join}\} \big) \\
\leq \quad & \sum_{S \in \mathcal{S}} p_S \big(c(S) + \min\{c(J) : J \text{ is a } T_S\text{-join}\} \big) \\
= \quad & c(x^*) + \sum_{S \in \mathcal{S}} p_S \min\{c(J) : J \text{ is a } T_S\text{-join}\} \\
\leq \quad & c(x^*) + \sum_{S \in \mathcal{S}} p_S c(y^S)
\end{aligned}
$$

for any set of correction vectors y^S ($S \in \mathcal{S}$) such that y^S is in the T_S-join polyhedron

$$
\left\{ y \in \mathbb{R}^E_{\geq 0} : y(C) \geq 1 \ \forall\ T_S\text{-cuts } C \right\}
$$

(Edmonds, Johnson [1973])

Fact: narrow cut C is a T_S-cut $\Leftrightarrow$ $|S \cap C|$ even

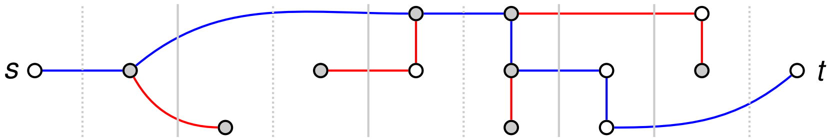
Correction vectors (An, Kleinberg, Shmoys [2012], Sebő [2013], V. [2015])

- Need a correction vector y^S with $y^S(C) \geq 1$ for all T_S-cuts C.
- $\frac{x^*}{2}$ is a valid correction vector except for narrow cuts C with $|S \cap C|$ even.
- In particular, it is valid for Gao trees.

Correction vectors

(An, Kleinberg, Shmoys [2012], Sebő [2013], V. [2015])

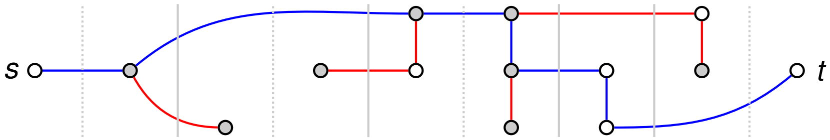
- Need a correction vector y^S with $y^S(C) \geq 1$ for all T_S-cuts C.
- $\frac{x^*}{2}$ is a valid correction vector except for narrow cuts C with $|S \cap C|$ even.
- In particular, it is valid for Gao trees.
- Let $S = I_S \dot\cup J_S$, where I_S is the s-t-path and J_S is the T_S-join.
- Then χ^{J_S} is a valid correction vector.



$S = I_S \dot\cup J_S$. Narrow cuts (grey) that need parity correction (solid)

Correction vectors (An, Kleinberg, Shmoys [2012], Sebő [2013], V. [2015])

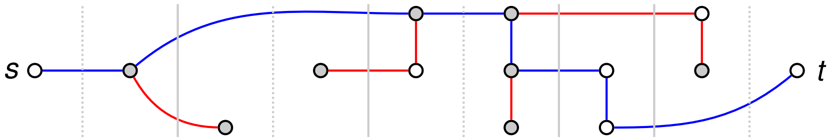
- Need a correction vector y^S with $y^S(C) \geq 1$ for all T_S-cuts C.
- $\frac{x^*}{2}$ is a valid correction vector except for narrow cuts C with $|S \cap C|$ even.
- In particular, it is valid for Gao trees.
- Let $S = I_S \dot\cup J_S$, where I_S is the s-t-path and J_S is the T_S-join.
- Then χ^{J_S} is a valid correction vector.
- Take a convex combination of $\frac{x^*}{2}$ and χ^{J_S} and fix violated narrow cuts by adding fractions of χ^{I_S}



$S = I_S \dot\cup J_S$. Narrow cuts (grey) that need parity correction (solid) contain (at least) one red and one blue edge. Thus $y^S = \frac{2}{3}\frac{x^*}{2} + \frac{1}{3}\chi^{J_S} + \frac{1}{3}\chi^{I_S}$ is valid

Correction vectors (An, Kleinberg, Shmoys [2012], Sebő [2013], V. [2015])

- Need a correction vector y^S with $y^S(C) \geq 1$ for all T_S-cuts C.
- $\frac{x^*}{2}$ is a valid correction vector except for narrow cuts C with $|S \cap C|$ even.
- In particular, it is valid for Gao trees.
- Let $S = I_S \,\dot\cup\, J_S$, where I_S is the s-t-path and J_S is the T_S-join.
- Then χ^{J_S} is a valid correction vector.
- Take a convex combination of $\frac{x^*}{2}$ and χ^{J_S} and fix violated narrow cuts by adding fractions of $\chi^{I_{S'}}$ for $S' \in \mathcal{S}$



$S = I_S \,\dot\cup\, J_S$. Narrow cuts (grey) that need parity correction (solid) contain (at least) one red and one blue edge. Thus $y^S = \frac{2}{3}\frac{x^*}{2} + \frac{1}{3}\chi^{J_S} + \frac{1}{3}\chi^{I_S}$ is valid

A good distribution

(Gottschalk, V. [2016])

- Gao trees do not need fixing.
- Unfortunately, there may be no cheap Gao tree. (Gao [2015])
- But the path l_S of a Gao tree can be used to help other trees.

A good distribution

(Gottschalk, V. [2016])

- Gao trees do not need fixing.
- Unfortunately, there may be no cheap Gao tree. (Gao [2015])
- But the path I_S of a Gao tree can be used to help other trees.

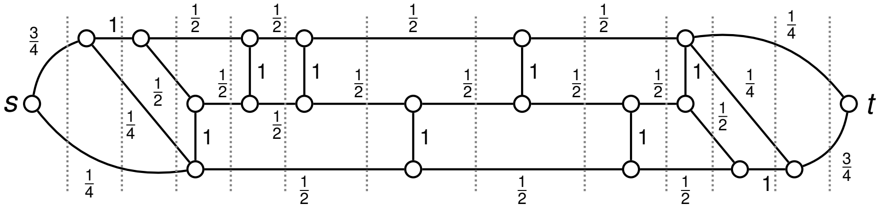
Main Theorem:
There are $S_1, \ldots, S_r \in \mathcal{S}$ and $p_1, \ldots, p_r > 0$ with $\sum_{j=1}^{r} p_j = 1$ and $x^* = \sum_{j=1}^{r} p_j \chi^{S_j}$, and for every narrow cut C there exists a k with $\sum_{j=1}^{k} p_j \geq 2 - x^*(C)$ and $|C \cap S_j| = 1$ for all $j = 1, \ldots, k$.

A good distribution

(Gottschalk, V. [2016])

- Gao trees do not need fixing.
- Unfortunately, there may be no cheap Gao tree. (Gao [2015])
- But the path l_S of a Gao tree can be used to help other trees.

Main Theorem:
There are $S_1, \ldots, S_r \in \mathcal{S}$ and $p_1, \ldots, p_r > 0$ with $\sum_{j=1}^{r} p_j = 1$ and $x^* = \sum_{j=1}^{r} p_j \chi^{S_j}$, and for every narrow cut C there exists a k with $\sum_{j=1}^{k} p_j \geq 2 - x^*(C)$ and $|C \cap S_j| = 1$ for all $j = 1, \ldots, k$.

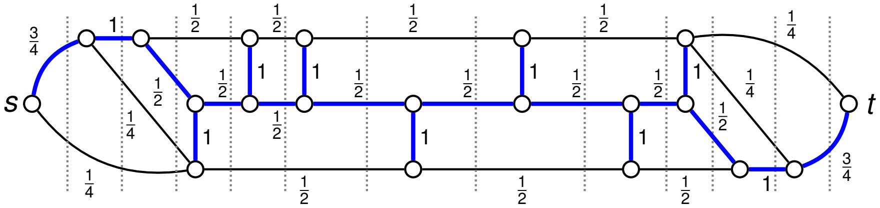


A good distribution

(Gottschalk, V. [2016])

- Gao trees do not need fixing.
- Unfortunately, there may be no cheap Gao tree. (Gao [2015])
- But the path I_S of a Gao tree can be used to help other trees.

Main Theorem:
There are $S_1, \ldots, S_r \in \mathcal{S}$ and $p_1, \ldots, p_r > 0$ with $\sum_{j=1}^r p_j = 1$ and $x^* = \sum_{j=1}^r p_j \chi^{S_j}$, and for every narrow cut C there exists a k with $\sum_{j=1}^k p_j \geq 2 - x^*(C)$ and $|C \cap S_j| = 1$ for all $j = 1, \ldots, k$.



S_1 with $p_1 = \frac{1}{2}$

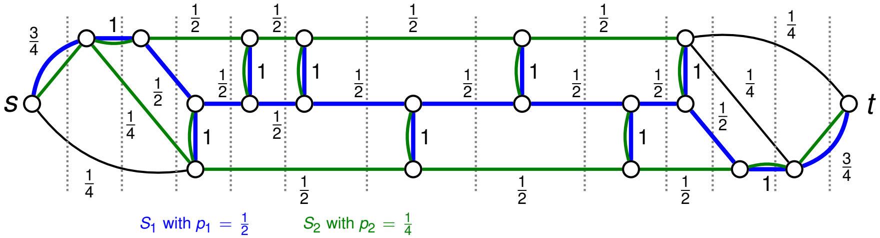
A good distribution

(Gottschalk, V. [2016])

- Gao trees do not need fixing.
- Unfortunately, there may be no cheap Gao tree. (Gao [2015])
- But the path I_S of a Gao tree can be used to help other trees.

Main Theorem:

There are $S_1, \dots, S_r \in \mathcal{S}$ and $p_1, \dots, p_r > 0$ with $\sum_{j=1}^{r} p_j = 1$ and $x^* = \sum_{j=1}^{r} p_j \chi^{S_j}$, and for every narrow cut C there exists a k with $\sum_{j=1}^{k} p_j \geq 2 - x^*(C)$ and $|C \cap S_j| = 1$ for all $j = 1, \dots, k$.

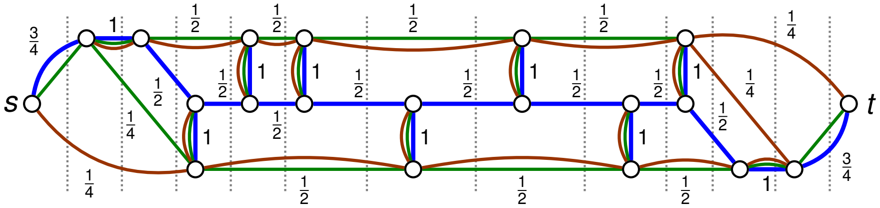


A good distribution

(Gottschalk, V. [2016])

- Gao trees do not need fixing.
- Unfortunately, there may be no cheap Gao tree. (Gao [2015])
- But the path I_S of a Gao tree can be used to help other trees.

<u>Main Theorem:</u>
There are $S_1, \ldots, S_r \in \mathcal{S}$ and $p_1, \ldots, p_r > 0$ with $\sum_{j=1}^{r} p_j = 1$ and $x^* = \sum_{j=1}^{r} p_j \chi^{S_j}$, and for every narrow cut C there exists a k with $\sum_{j=1}^{k} p_j \geq 2 - x^*(C)$ and $|C \cap S_j| = 1$ for all $j = 1, \ldots, k$.

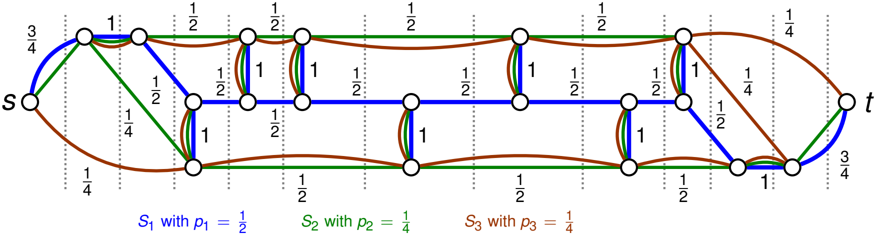


S_1 with $p_1 = \frac{1}{2}$ S_2 with $p_2 = \frac{1}{4}$ S_3 with $p_3 = \frac{1}{4}$

A good distribution

(Gottschalk, V. [2016])

<u>Main Theorem:</u>
There are $S_1, \ldots, S_r \in \mathcal{S}$ and $p_1, \ldots, p_r > 0$ with $\sum_{j=1}^r p_j = 1$ and $x^* = \sum_{j=1}^r p_j \chi^{S_j}$, and for every narrow cut C there exists a k with $\sum_{j=1}^k p_j \geq 2 - x^*(C)$ and $|C \cap S_j| = 1$ for all $j = 1, \ldots, k$.

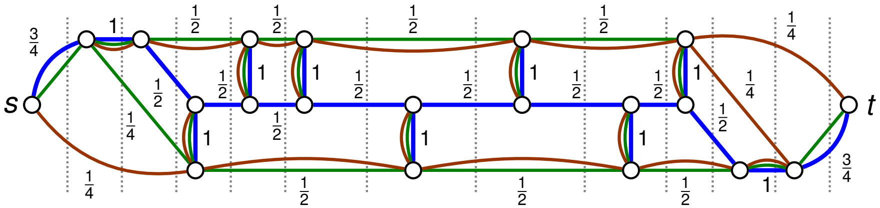


- implies Gao's Theorem (take S_1)

A good distribution

(Gottschalk, V. [2016])

<u>Main Theorem:</u>
There are $S_1, \ldots, S_r \in \mathcal{S}$ and $p_1, \ldots, p_r > 0$ with $\sum_{j=1}^r p_j = 1$ and $x^* = \sum_{j=1}^r p_j \chi^{S_j}$, and for every narrow cut C there exists a k with $\sum_{j=1}^k p_j \geq 2 - x^*(C)$ and $|C \cap S_j| = 1$ for all $j = 1, \ldots, k$.



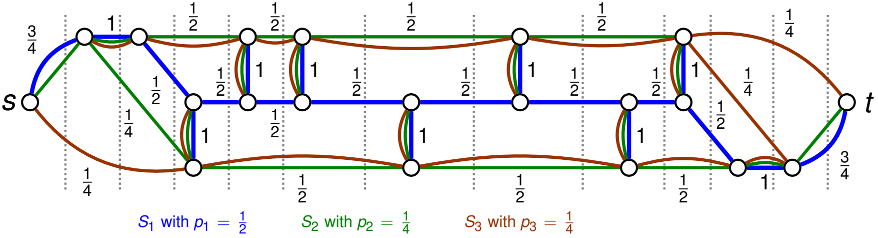
S_1 with $p_1 = \frac{1}{2}$ S_2 with $p_2 = \frac{1}{4}$ S_3 with $p_3 = \frac{1}{4}$

- implies Gao's Theorem (take S_1)
- can be computed in polynomial time

A good distribution

(Gottschalk, V. [2016])

<u>Main Theorem:</u>
There are $S_1, \ldots, S_r \in \mathcal{S}$ and $p_1, \ldots, p_r > 0$ with $\sum_{j=1}^r p_j = 1$ and $x^* = \sum_{j=1}^r p_j \chi^{S_j}$, and for every narrow cut C there exists a k with $\sum_{j=1}^k p_j \geq 2 - x^*(C)$ and $|C \cap S_j| = 1$ for all $j = 1, \ldots, k$.

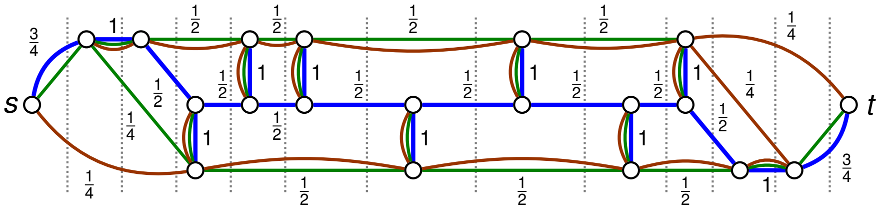


- implies Gao's Theorem (take S_1)
- can be computed in polynomial time
- yields approximation ratio 1.566 (with best-of-many)

A good distribution

(Gottschalk, V. [2016])

<u>Main Theorem:</u>
There are $S_1, \ldots, S_r \in \mathcal{S}$ and $p_1, \ldots, p_r > 0$ with $\sum_{j=1}^r p_j = 1$ and $x^* = \sum_{j=1}^r p_j \chi^{S_j}$, and for every narrow cut C there exists a k with $\sum_{j=1}^k p_j \geq 2 - x^*(C)$ and $|C \cap S_j| = 1$ for all $j = 1, \ldots, k$.



S_1 with $p_1 = \frac{1}{2}$ S_2 with $p_2 = \frac{1}{4}$ S_3 with $p_3 = \frac{1}{4}$

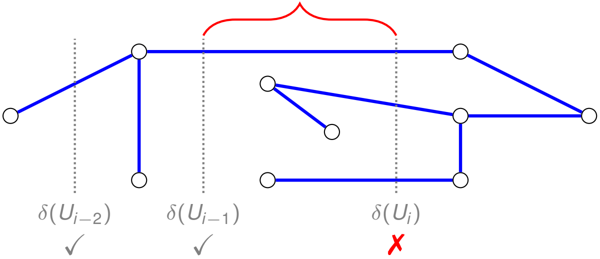
- implies Gao's Theorem (take S_1)
- can be computed in polynomial time
- yields approximation ratio 1.566 (with best-of-many)
- also used by Sebő and van Zuylen [2016] for ratio $\frac{26}{17}$

Proof: outline

- Start with
$$x^* = \frac{1}{r}\sum_{j=1}^{r}\chi^{S_j}$$
- Deal with the trees $S_j \ (j = 1, \ldots, r)$ in this order
- For each j: let
$$\{s\} = U_1 \subset \cdots \subset U_k = V \setminus \{t\}$$
be the sets with
$$x^*(\delta(U_i)) \leq 2 - \frac{j}{r}$$
- Deal with the cuts $\delta(U_i) \ (i = 1, \ldots, k)$ in this order
- Need $|S_j \cap \delta(U_i)| = 1$
- First make $S_j[U_i]$ connected

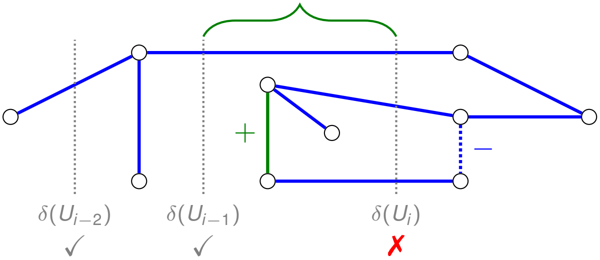
Proof: make $S_j[U_i]$ connected

Case 1: $S_j[U_i \setminus U_{i-1}]$ disconnected



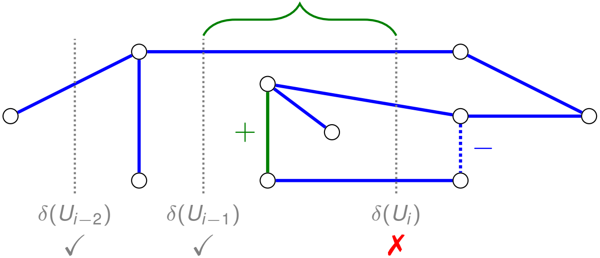
Proof: make $S_j[U_i]$ connected

Case 1: $S_j[U_i \setminus U_{i-1}]$ disconnected $\Rightarrow \exists\, k > j : S_k[U_i \setminus U_{i-1}]$ connected

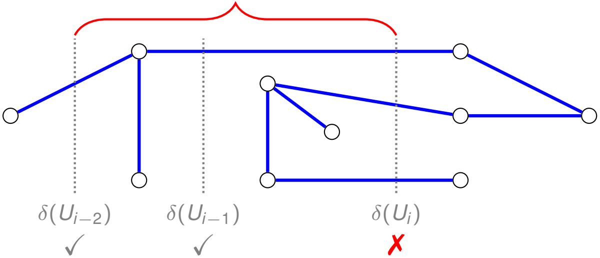


Proof: make $S_j[U_i]$ connected

Case 1: $S_j[U_i \setminus U_{i-1}]$ disconnected $\Rightarrow \exists\, k > j : S_k[U_i \setminus U_{i-1}]$ connected

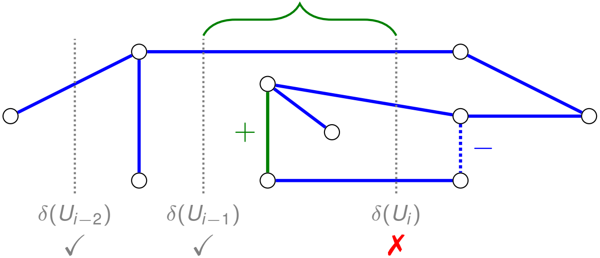


Case 2: $S_j[U_i \setminus U_{i-1}]$ connected

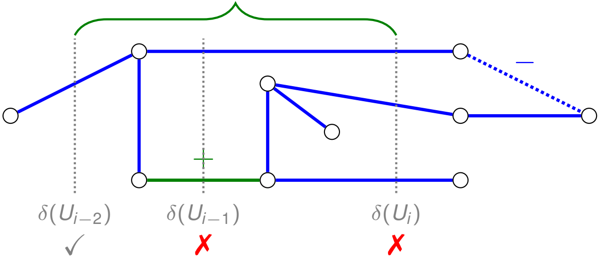


Proof: make $S_j[U_i]$ connected

Case 1: $S_j[U_i \setminus U_{i-1}]$ disconnected $\Rightarrow \exists\, k > j : S_k[U_i \setminus U_{i-1}]$ connected

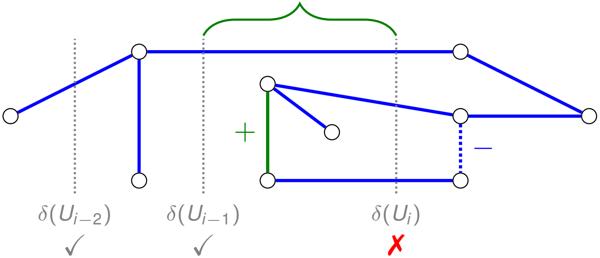


Case 2: $S_j[U_i \setminus U_{i-1}]$ connected $\Rightarrow \exists\, k > j : S_k \cap \delta(U_{i-1}) \cap \delta(U_i) = \emptyset$

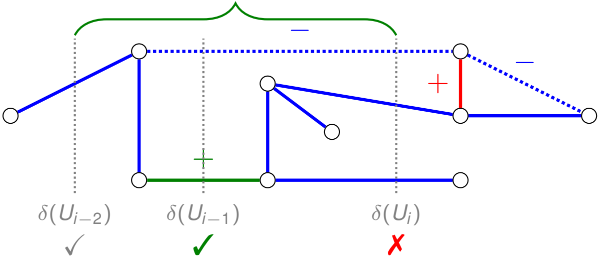


Proof: make $S_j[U_i]$ connected

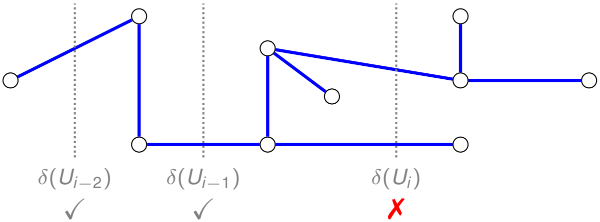
Case 1: $S_j[U_i \setminus U_{i-1}]$ disconnected $\Rightarrow \exists\, k > j : S_k[U_i \setminus U_{i-1}]$ connected



Case 2: $S_j[U_i \setminus U_{i-1}]$ connected $\Rightarrow \exists\, k > j : S_k \cap \delta(U_{i-1}) \cap \delta(U_i) = \emptyset$

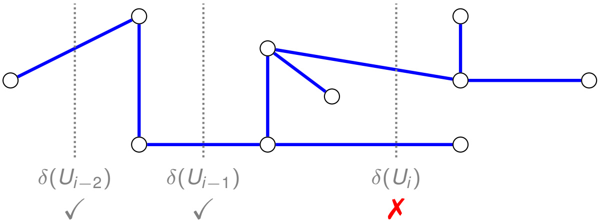


Proof: if $S_j[U_i]$ connected, get $|S_j \cap \delta(U_i)| = 1$



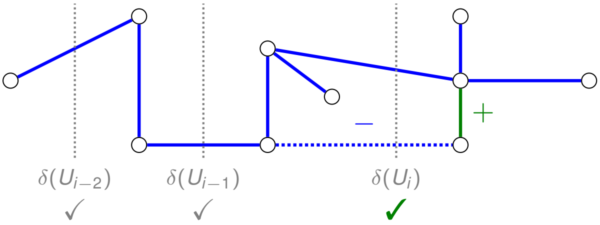
Proof: if $S_j[U_i]$ connected, get $|S_j \cap \delta(U_i)| = 1$

Then $\exists\, k > j : |S_k \cap \delta(U_i)| = 1$.



Proof: if $S_j[U_i]$ connected, get $|S_j \cap \delta(U_i)| = 1$

Then $\exists\, k > j : |S_k \cap \delta(U_i)| = 1$.



TSP variants – state of the art

Integrality ratios. Upper bounds = approximation ratios unless mentioned otherwise

2ECSS, general metrics:

- between $\frac{6}{5}$ and $\frac{3}{2}$ (Alexander, Boyd, Elliott-Magwood [2006])

2ECSS, unweighted graphs:

- between $\frac{8}{7}$ (Boyd, Fu, Sun [2014]) and $\frac{4}{3}$ (Sebő, V. [2014])

TSP, general metrics:

- between $\frac{4}{3}$ and $\frac{3}{2}$ (Wolsey [1980])

TSP, unweighted graphs:

- between $\frac{4}{3}$ and $\frac{7}{5}$ (Sebő, V. [2014])

s-t-path TSP, general metrics:

- between $\frac{3}{2}$ and $\frac{26}{17}$ (Sebő, van Zuylen [2016])

s-t-path TSP, unweighted graphs:

- $\frac{3}{2}$ (Sebő, V. [2014])

ATSP, $\triangle$-inequality:

- between 2 (Boyd, Elliott-Magwood [2005], Charikar, Goemans, Karloff [2006])
  and $\log^{O(1)} \log n$ (Anari and Oveis Gharan [2015]);
  apx ratio $8 \log n / \log \log n$ (Asadpour, Goemans, Mądry, Oveis Gharan, Saberi [2010])

ATSP, unweighted digraphs:

- between $\frac{3}{2}$ (Gottschalk [2013]) and 13; apx ratio $27 + \epsilon$ (Svensson [2015])

How important are integrality ratios?

- We cannot solve the LPs combinatorially in polynomial time.
- Integrality ratios do not imply lower bounds on approximability.

Example: Euclidean TSP

- approximation scheme (Arora [1998])
- subtour relaxation has integrality ratio $\frac{4}{3}$ (Hougardy [2014])

- Integrality ratios imply bounds on what we can achieve if we use this LP as lower bound.

Current and future research

- better than $\frac{3}{2}$ for s-t-path TSP in graphs?
- $\frac{3}{2}$ for s-t-path TSP in general metrics?
- better than Sebő's $\frac{8}{5}$ for T-tours in general metrics?
- improve on Christofides' $\frac{3}{2}$-approximation algorithm? $\frac{4}{3}$?
- constant factor for asymmetric TSP?
- generalizations and practical applications

Thank you!

- J. Vygen: New approximation algorithms for the TSP.
  OPTIMA 90 (2012), 1–12
- A. Sebő, J. Vygen: Shorter tours by nicer ears: 7/5-approximation for the graph-TSP, 3/2 for the path version, and 4/3 for two-edge-connected subgraphs.
  Combinatorica 34 (2014), 597–629
- J. Vygen: Reassembling trees for the traveling salesman.
  SIAM Journal on Discrete Mathematics 30 (2016), 875–894
- C. Gottschalk, J. Vygen: Better *s*-*t*-tours by Gao trees.
  Proceedings of IPCO 2016, 126–137