s-t path TSP and the randomized Christofides algorithm

Shatian Wang

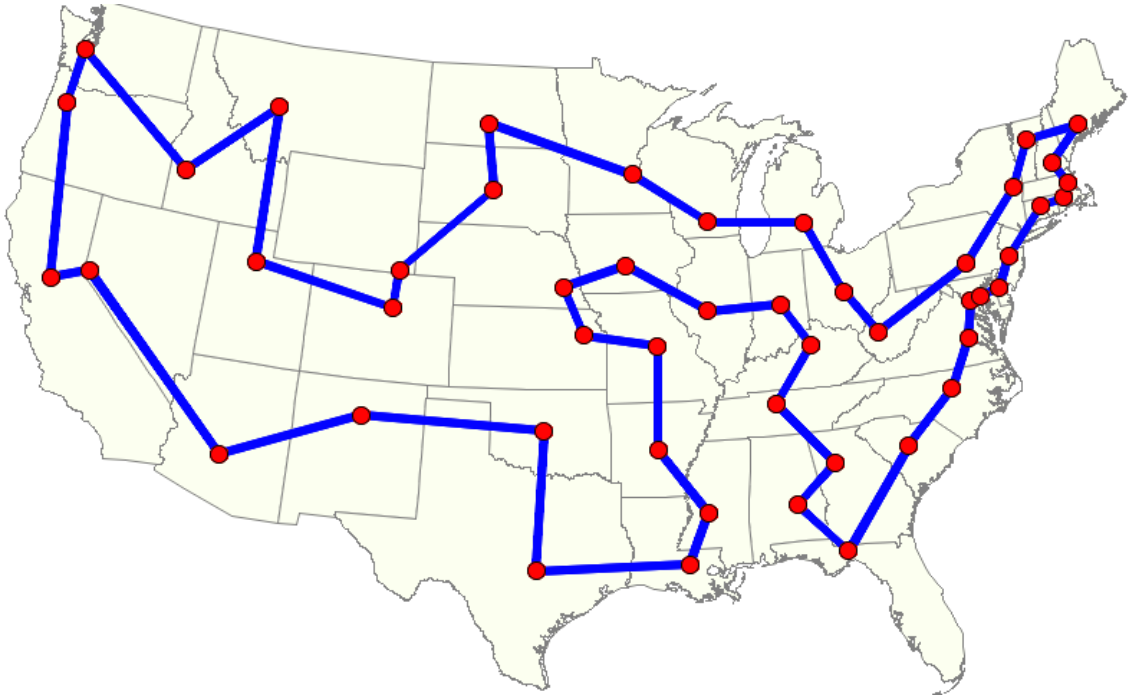
URA @ CO, UWaterloo

Oct., 2016

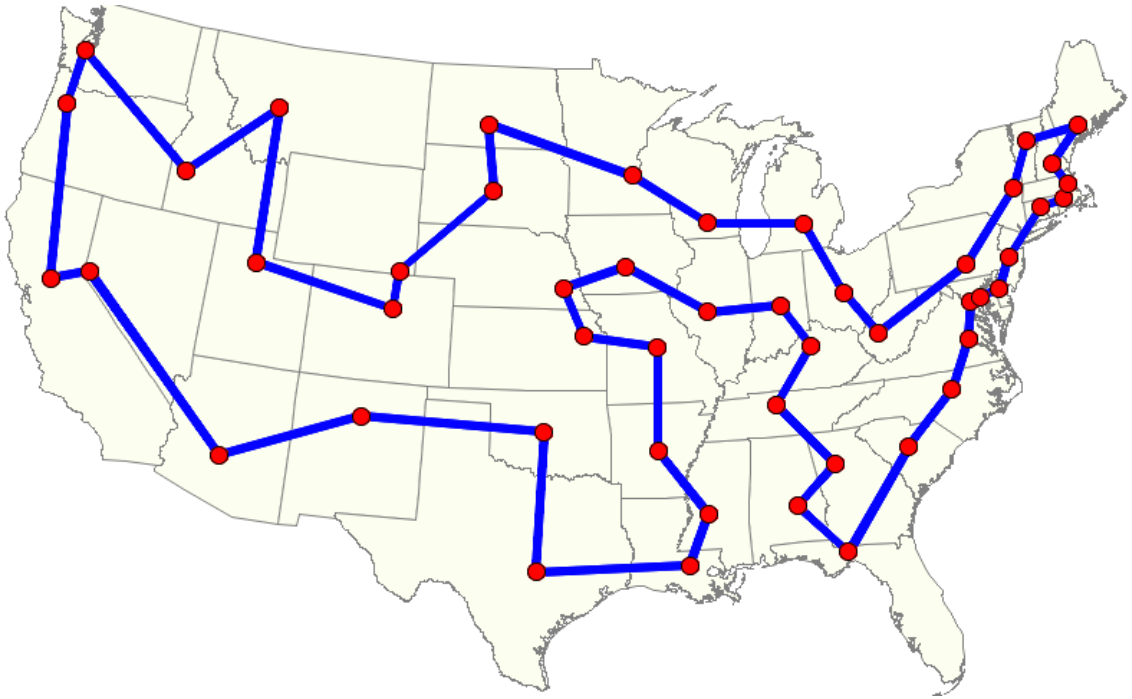
Combinatorial Optimization

- Finite set of objects S
- Optimal object $o_{opt} \in S$
- Exhaustive search?

TSP: 49-City Instance



TSP: 49-City Instance



$|S| = (49\text{-}1)!/2 =$

6,206,957,796,268,036,335,431,144,523,686,
687,519,260,743,177,338,880,000,000,000

General TSP

- NP-hard
  - **no** polynomial time algorithm that finds the **exact optimal solution**: **OPT**.

General TSP

- NP-hard
  - **no** polynomial time algorithm that finds the **exact optimal solution**: **OPT**.
- Approximation algorithm
  - algorithm that finds a **near to optimal solution**.

General TSP

- NP-hard
  - **no** polynomial time algorithm that finds the **exact optimal solution**: **OPT**.
- Approximation algorithm
  - algorithm that finds a **near to optimal solution**.
- Minimization Problem (e.g. TSP)
  - assign a cost to each object
  - optimal object minimize the cost

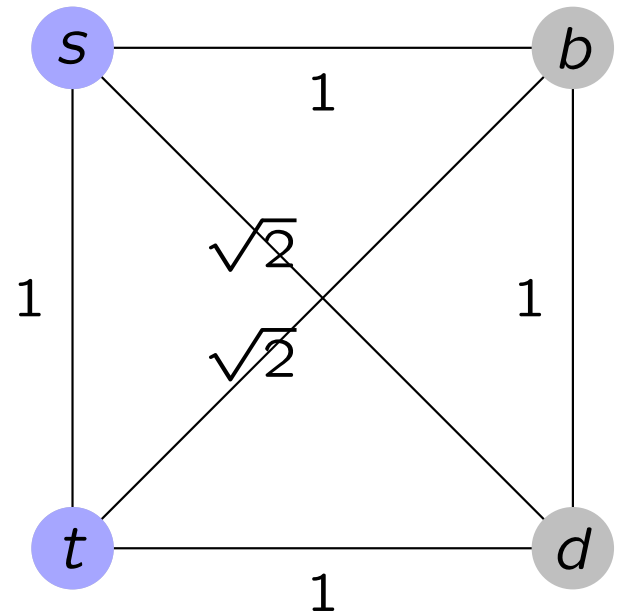
General TSP

- NP-hard
  - **no** polynomial time algorithm that finds the **exact optimal solution**: **OPT**.
- Approximation algorithm
  - algorithm that finds a **near to optimal solution**.
- Minimization Problem (e.g. TSP)
  - assign a cost to each object
  - optimal object minimize the cost
- Approximation ratio α
  - **c(output)** $\leq \alpha \times$ **c(OPT)**.
  - α-approximation algorithm.

metric s-t path TSP

- Given a **complete graph** $G = (V, E)$ with **metric** edge costs and two fixed vertices $\mathbf{s, t \in V}$.

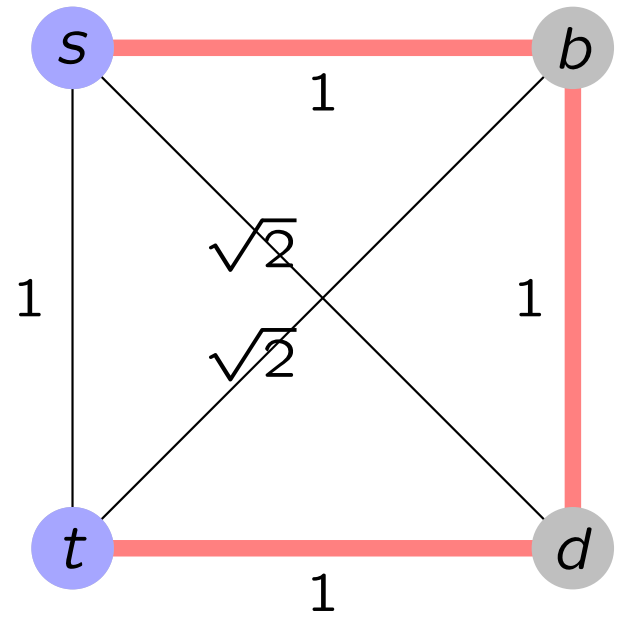
- Find a minimum cost **Hamiltonian path from s to t**.
  - a path that meets every graph vertex exactly once
  - starting from s and ending at t



metric s-t path TSP

- Given a **complete graph** $G = (V, E)$ with **metric** edge costs and two fixed vertices $\mathbf{s, t \in V}$.

- Find a minimum cost **Hamiltonian path from s to t**.
  - a path that meets every graph vertex exactly once
  - starting from s and ending at t



Christofides' Algorithm (Extended by Hoogeveen 1990)

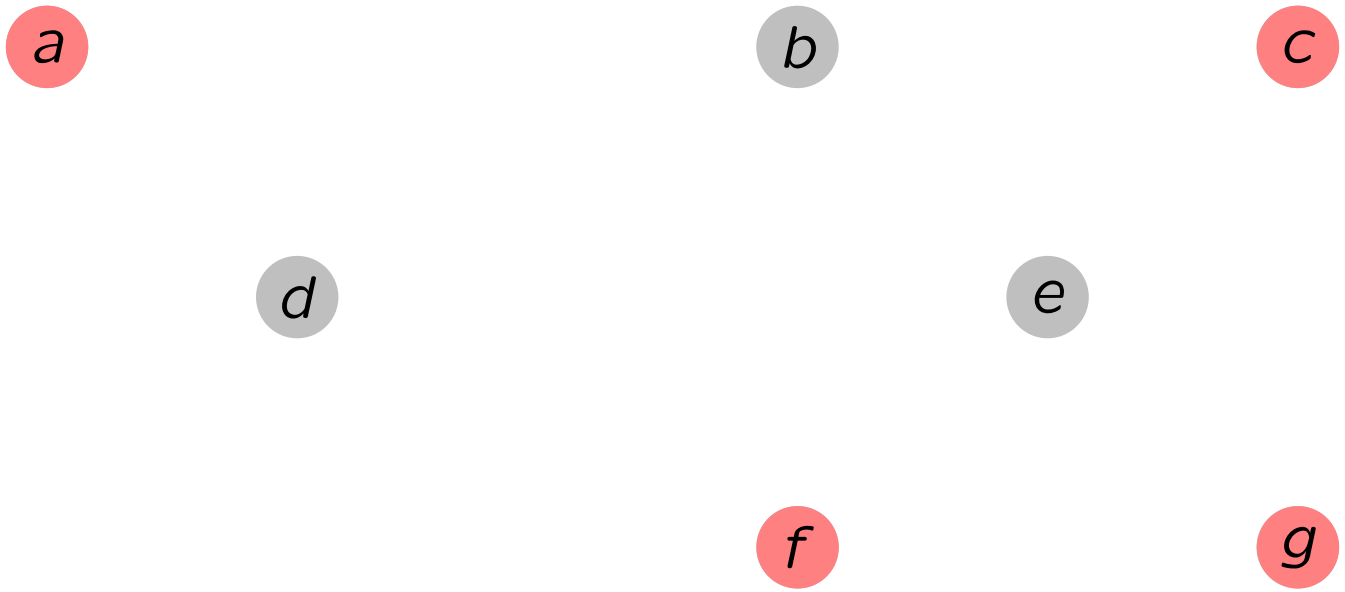
Definition (**T-join**)

T := set of **vertices**; |T| = even
F := set of **edges**, s.t. $dgr(v)$ = odd w.r.t. $F \Leftrightarrow v \in T$.
F is a **T-join**.

Christofides' Algorithm (Extended by Hoogeveen 1990)

Definition (**T-join**)

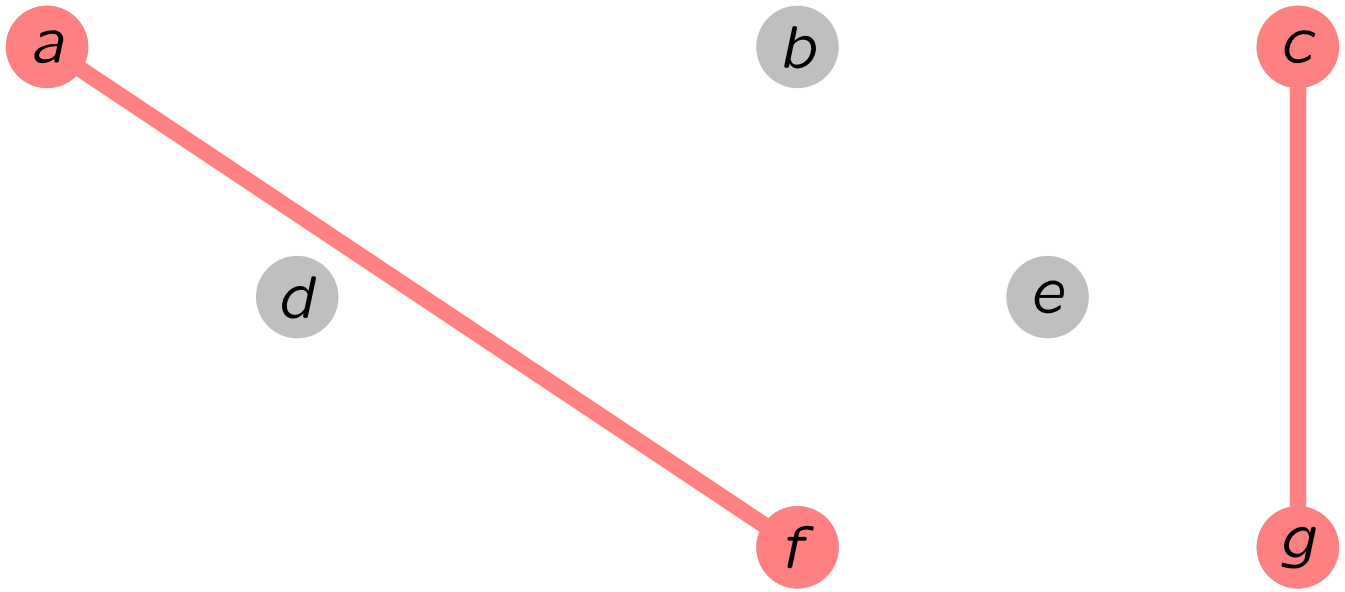
T := set of **vertices**; |T| = even
F := set of **edges**, s.t. $dgr(v)$ = odd w.r.t. $F \Leftrightarrow v \in T$.
F is a **T-join**.



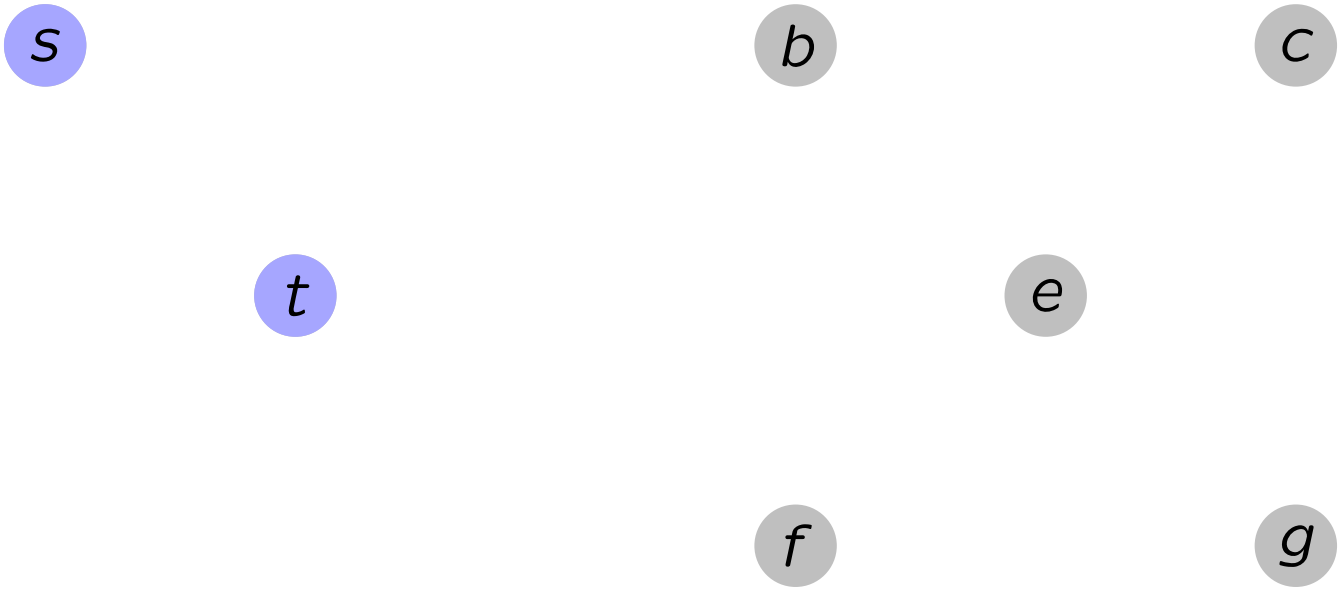
Christofides' Algorithm (Extended by Hoogeveen 1990)

Definition (**T-join**)

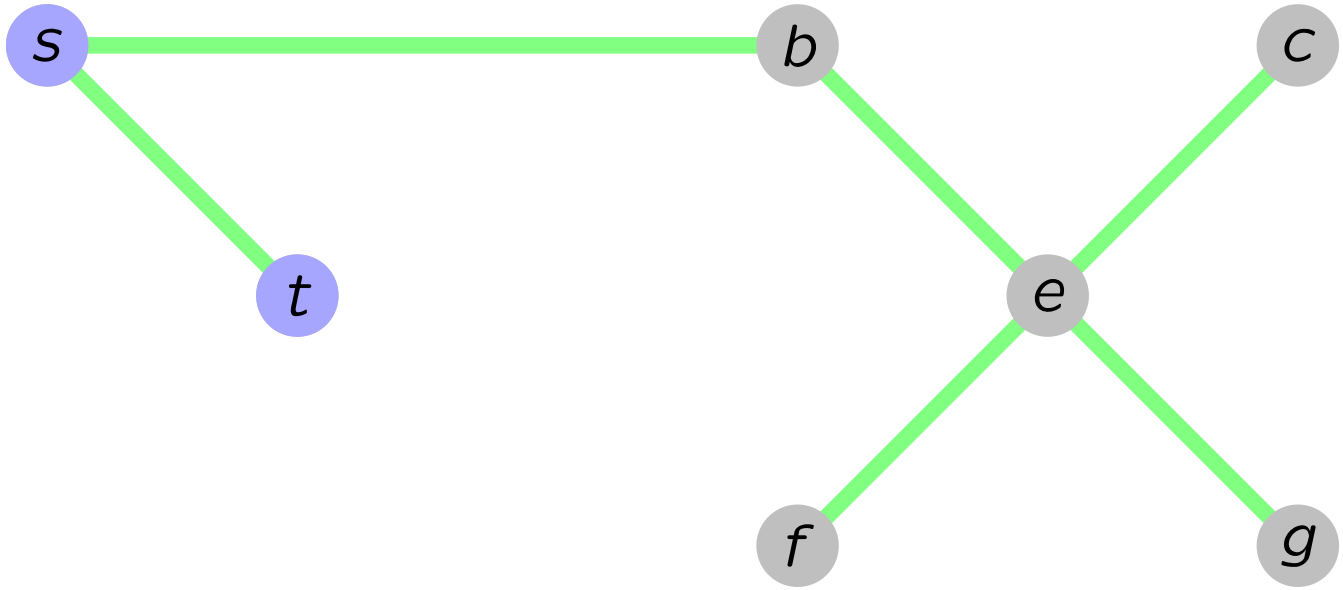
T := set of **vertices**; |T| = even
F := set of **edges**, s.t. $dgr(v)$ = odd w.r.t. $F \Leftrightarrow v \in T$.
F is a **T-join**.



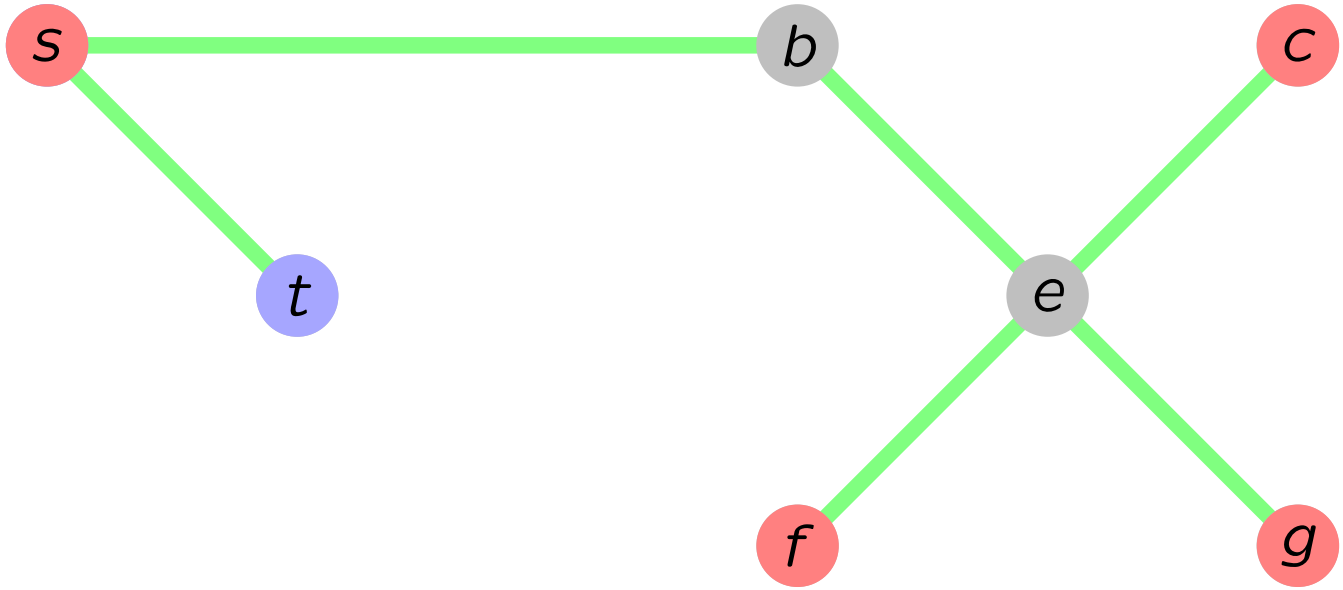
Christofides' Algorithm (Extended by Hoogeveen 1990)



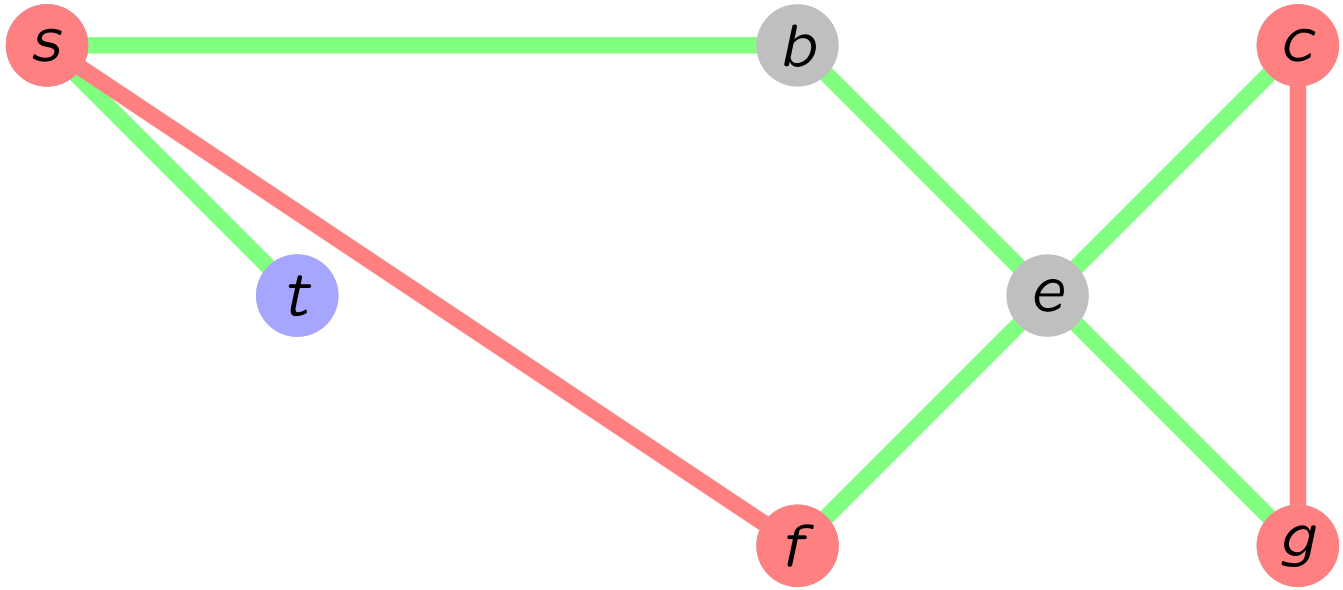
Christofides' Algorithm (Extended by Hoogeveen 1990)



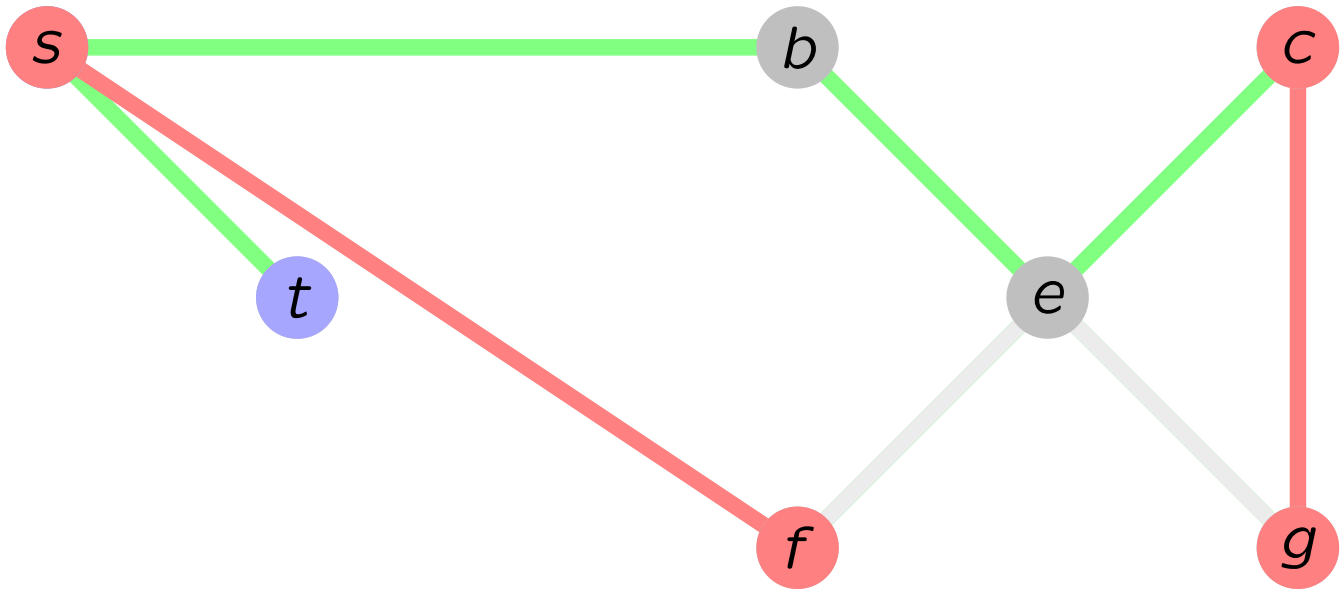
Christofides' Algorithm (Extended by Hoogeveen 1990)



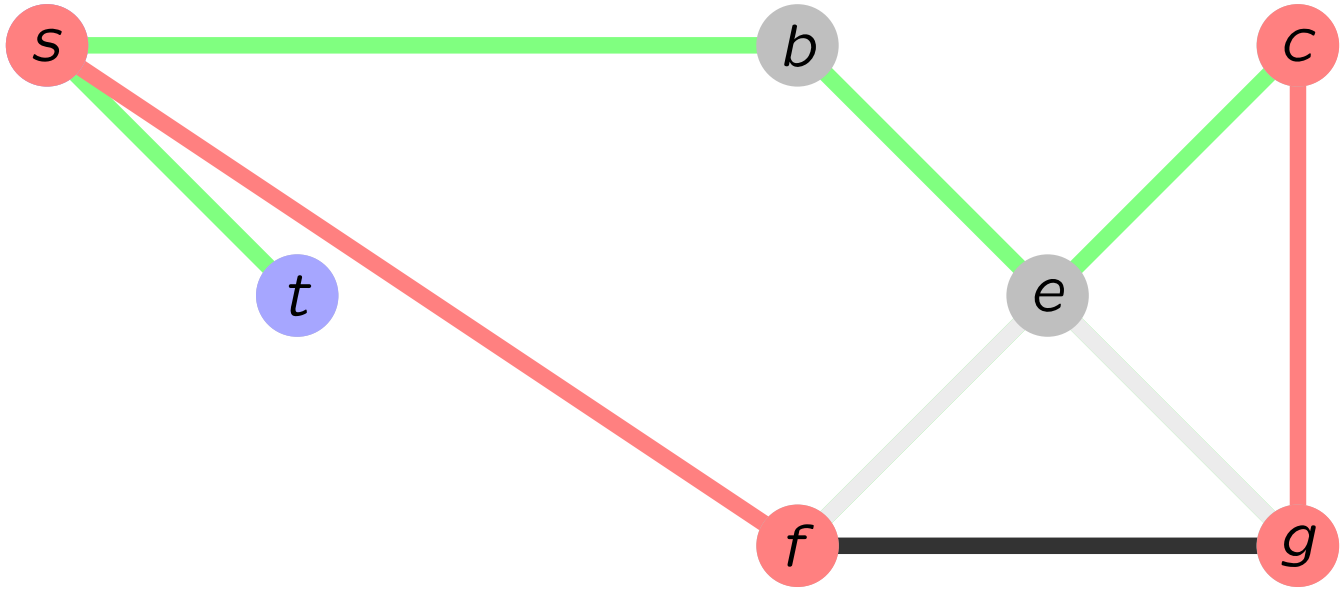
Christofides' Algorithm (Extended by Hoogeveen 1990)



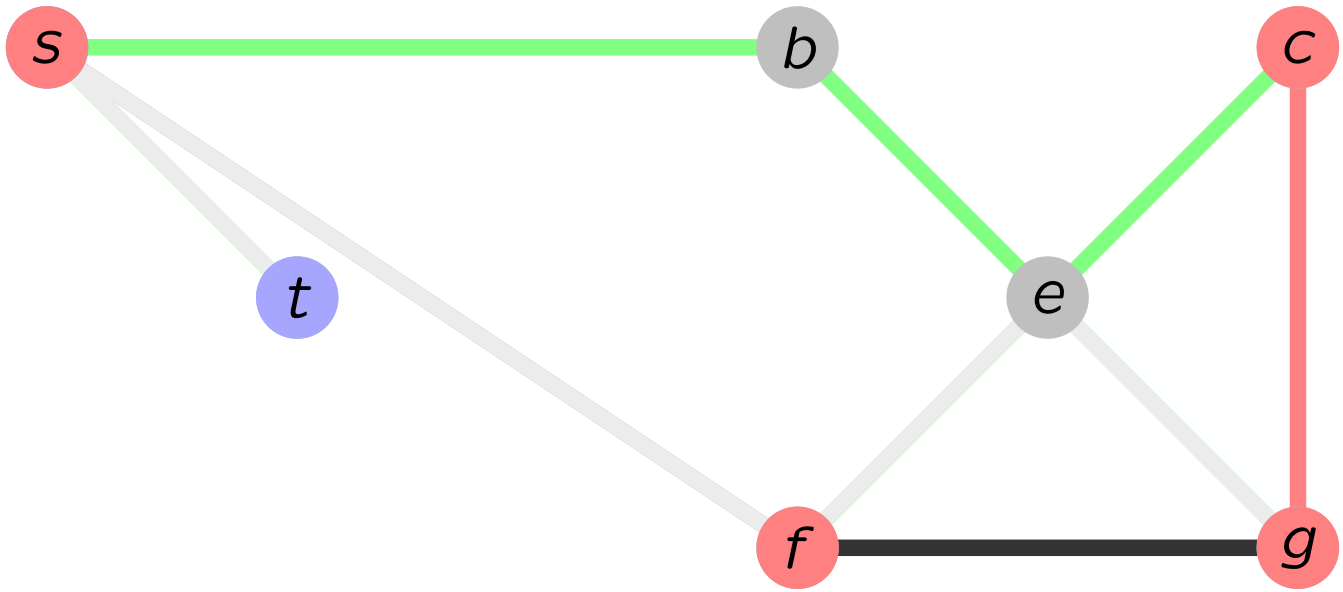
Christofides' Algorithm (Extended by Hoogeveen 1990)



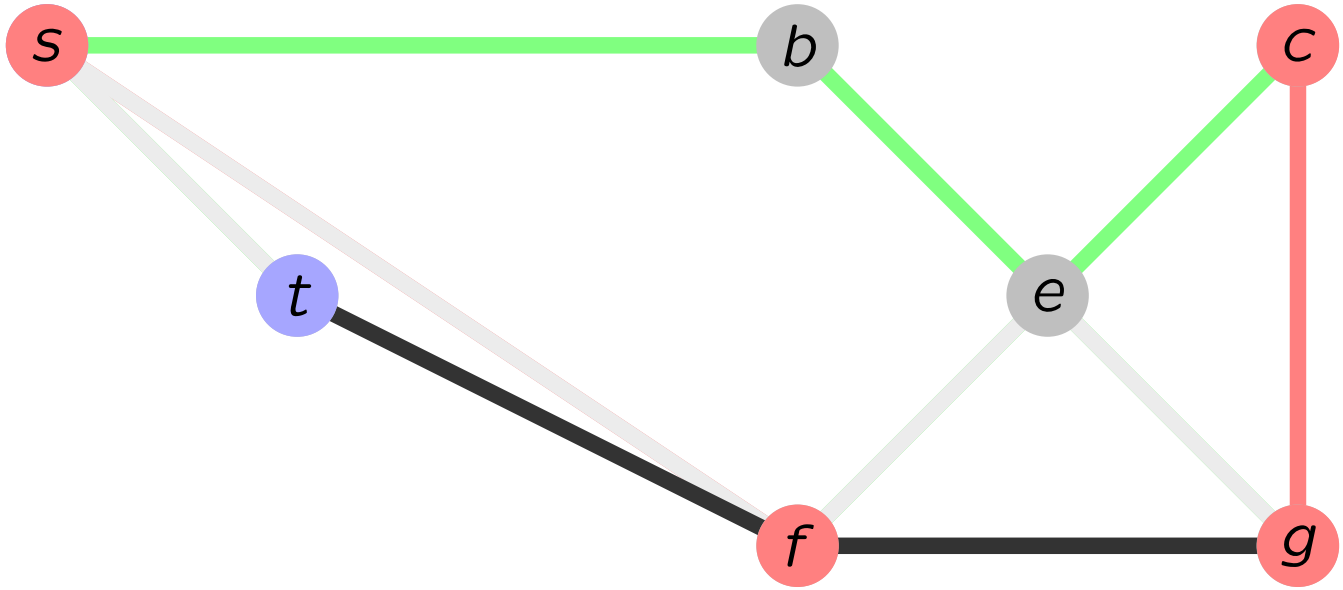
Christofides' Algorithm (Extended by Hoogeveen 1990)



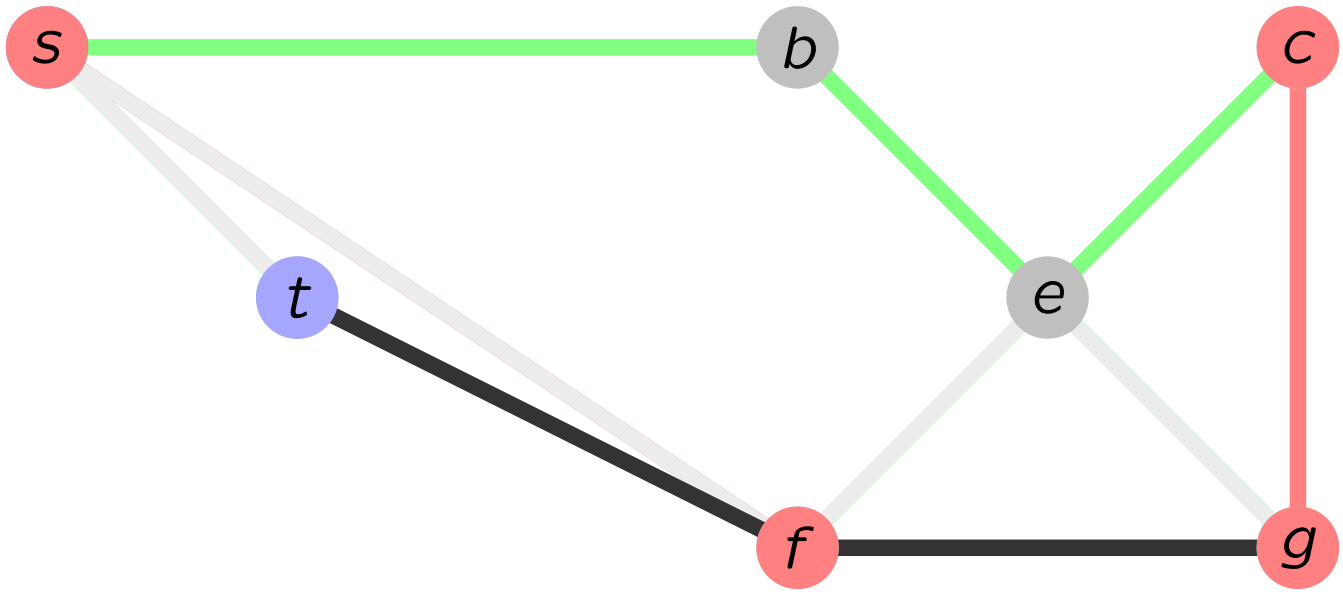
Christofides' Algorithm (Extended by Hoogeveen 1990)



Christofides' Algorithm (Extended by Hoogeveen 1990)

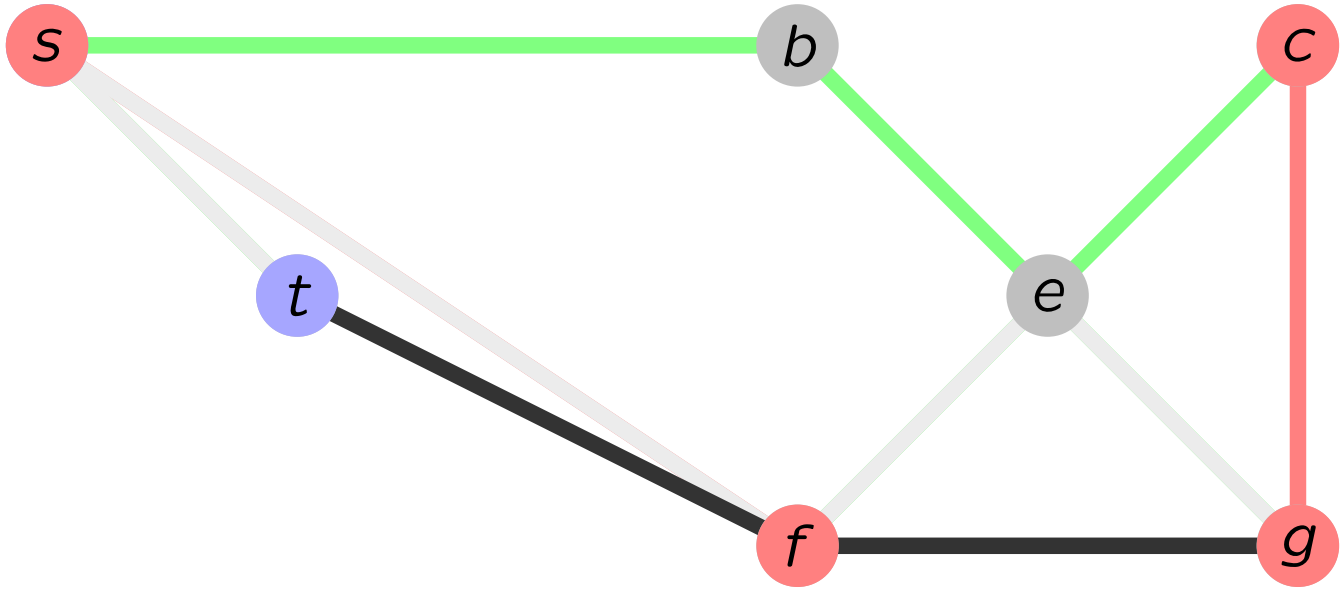


Christofides' Algorithm (Extended by Hoogeveen 1990)



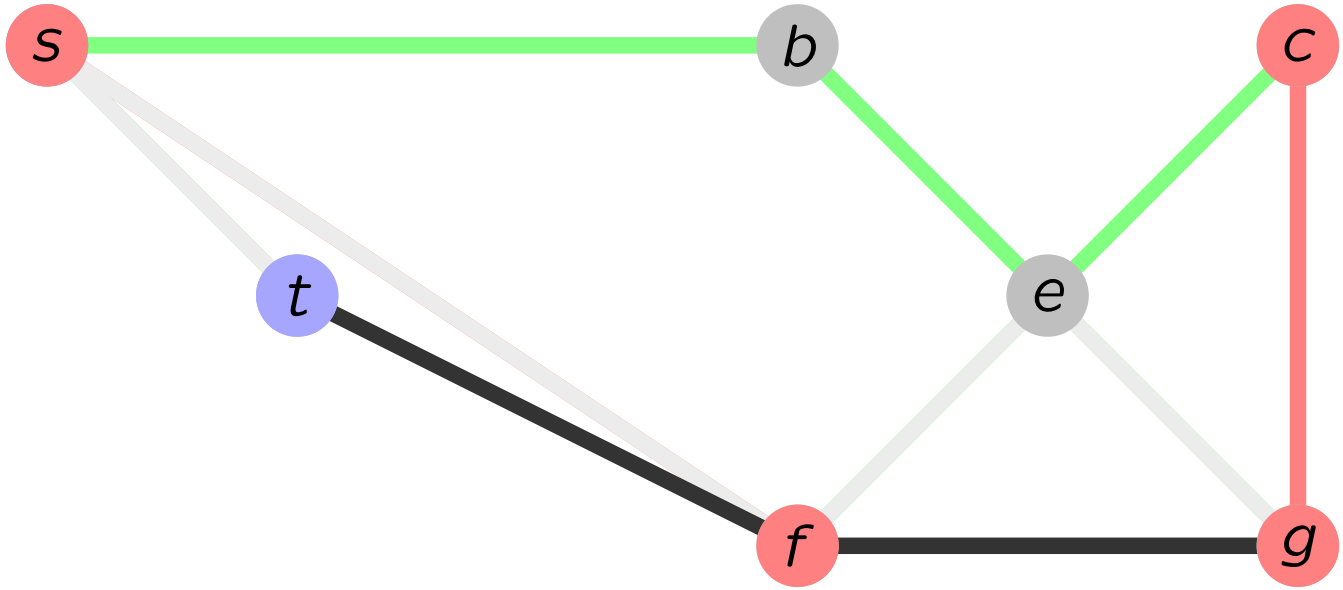
- $c(HamiltonianPath) \leq c(EularienWalk) = c(J + F) = c(J) + c(F)$

Christofides' Algorithm (Extended by Hoogeveen 1990)



- $c(HamiltonianPath) \leq c(EularienWalk) = c(J + F) = c(J) + c(F)$
- $c(J) \leq c(\mathbf{OPT})$; $c(F) \leq \frac{2}{3}c(\mathbf{OPT})$

Christofides' Algorithm (Extended by Hoogeveen 1990)



- $c(HamiltonianPath) \leq c(EularienWalk) = c(J + F) = c(J) + c(F)$
- $c(J) \leq c(\mathbf{OPT})$; $c(F) \leq \frac{2}{3}c(\mathbf{OPT})$
- **$\frac{5}{3}$-approximation algorithm**

An, Kleinberg and Shmoys (2012) the Randomized Christofides

- First improvement to $\frac{5}{3}$ (1990 Hoogeveen)
- LP-based algorithm:
  - the **"Randomized Christofides"**
- Upper bounded approximation ratio by
  - $\frac{\sqrt{5}+1}{2} = 1.618$

LP: Linear Programming

- Objective function to minimize/maximize
- Linear constraints

LP: Linear Programming

- Objective function to minimize/maximize
- Linear constraints
- Example:
- minimize: $c_1x_1 + c_2x_2$
- subject to:
  - $a_{11}x_1 + a_{12}x_2 \leq b_1$
  - $a_{21}x_1 + a_{22}x_2 \leq b_2$
  - $a_{31}x_1 + a_{32}x_2 \leq b_3$
  - $x_1 \geq 0$
  - $x_2 \geq 0$

LP: Linear Programming

- Objective function to minimize/maximize
- Linear constraints
- Example:

- minimize: $c_1x_1 + c_2x_2$
- subject to:
  - $a_{11}x_1 + a_{12}x_2 \leq b_1$
  - $a_{21}x_1 + a_{22}x_2 \leq b_2$
  - $a_{31}x_1 + a_{32}x_2 \leq b_3$
  - $x_1 \geq 0$
  - $x_2 \geq 0$

- minimize: $\mathbf{c^T x}$
- subject to:
  - $\mathbf{Ax \leq b}$
  - $\mathbf{x \geq 0}$
- where $\mathbf{c} = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix}$, $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$, $\mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$, $\mathbf{A} = \begin{bmatrix} a_{11} a_{12} a_{13} \\ a_{21} a_{22} a_{23} \\ a_{31} a_{32} a_{33} \end{bmatrix}$

s-t path TSP: IP Formulation

- minimize:

$$\sum_{e \in E} c_e x_e = \mathbf{c^T x}$$

- subject to:
  - $x(\delta(s)) = x(\delta(t)) = 1$
  - $x(\delta(v)) = 2$ ———————— $\forall\ v \neq s, t$
  - $x(\delta(S)) \geq 1$ ———————— $\forall\ |S \cap \{s, t\}| = 1$
  - $x(\delta(S)) \geq 2$ ———————— $\forall\ \emptyset \subsetneq S \subsetneq V$, $|S \cap \{s, t\}|$ even
  - $x_e = 0$ or $x_e = 1$ ———————— $\forall\ e \in E$
- **Also NP-hard!**

s-t path TSP: LP-relaxations

- **L.P.1 (Held-Karp Relaxation for s-t path TSP)**
  - minimize:
$$\sum_{e \in E} c_e x_e = \mathbf{c}^T \mathbf{x}$$
  - subject to:
    - $x(\delta(s)) = x(\delta(t)) = 1$
    - $x(\delta(v)) = 2$ ——————— $\forall\ v \neq s, t$
    - $x(\delta(S)) \geq 1$ ——————— $\forall\ |S \cap \{s, t\}| = 1$
    - $x(\delta(S)) \geq 2$ ——————— $\forall\ \emptyset \subsetneq S \subsetneq V$, $|S \cap \{s, t\}|$ even
    - $1 \geq x_e \geq 0$ ——————— $\forall\ e \in E$
- **Can compute optimal solution $\mathbf{x}^*$ in polytime.**
- **$\mathbf{x}^*$ might contain fractional values.**

The Randomized Christofides Algorithm

1. Find the optimal solution to LP1: $\mathbf{x}^*$
2. Write $\mathbf{x}^*$ as a convex combination of spanning trees:

$$\mathbf{x}^* = \sum p_i \chi^{J_i}$$

   - χ^{J_i} is the incidence vector of spanning tree J_i
   - $p_i \geq 0, \sum p_i = 1$

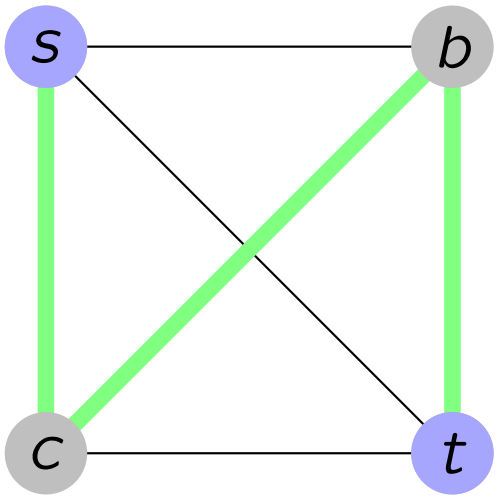
3. Sample a spanning tree $\mathbf{J}$ according to the probability defined by p_i's.
4. Perform Christofides algorithm on $\mathbf{J}$.

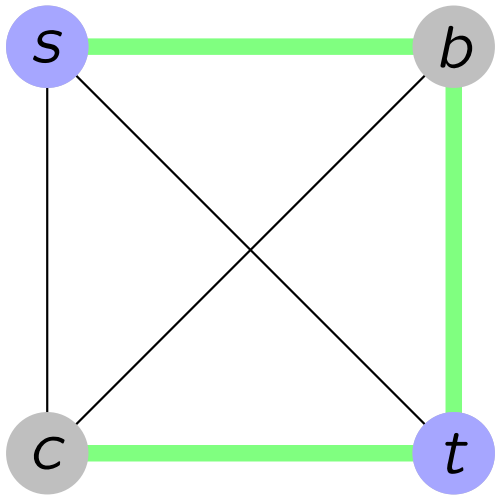
The Randomized Christofides: Made-up Example

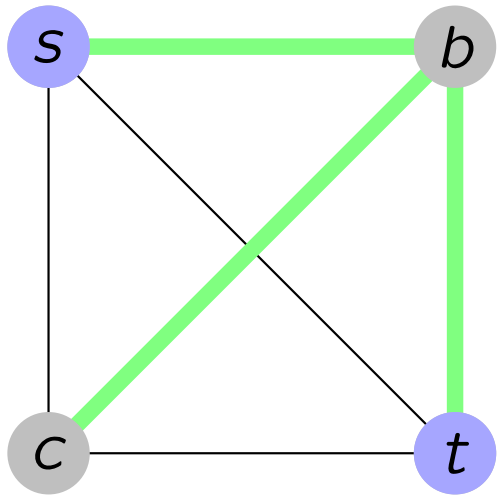
$\mathcal{X}^{J_1}$: $[1, 0, 1, 0, 1, 0]^T$

$\mathcal{X}^{J_2}$: $[0, 1, 1, 1, 0, 0]^T$

$\mathcal{X}^{J_3}$: $[0, 1, 1, 0, 1, 0]^T$



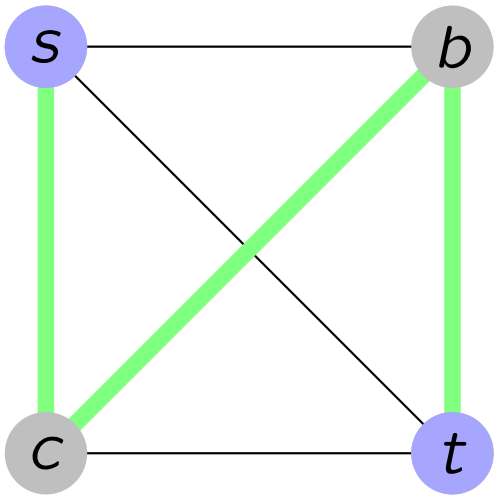




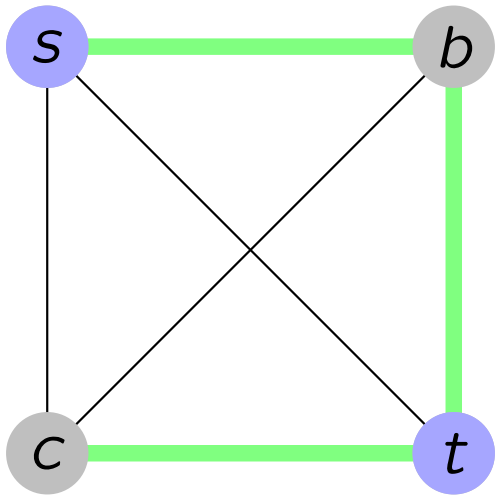
$$\mathbf{x}^* = 0.3\mathcal{X}^{J_1} + 0.2\mathcal{X}^{J_2} + 0.5\mathcal{X}^{J_3}$$

The Randomized Christofides: Made-up Example

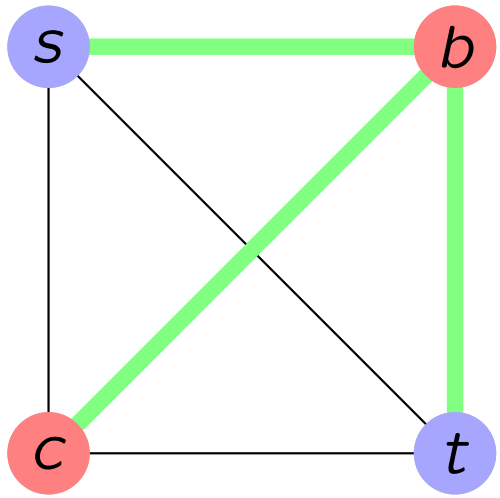
$\mathcal{X}^{J_1}$:
$[1, 0, 1, 0, 1, 0]^T$



$\mathcal{X}^{J_2}$:
$[0, 1, 1, 1, 0, 0]^T$



$\mathcal{X}^{J_3}$:
$[0, 1, 1, 0, 1, 0]^T$



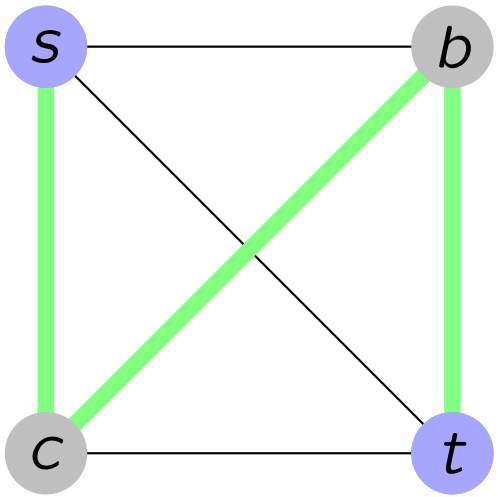
$$\mathbf{x}^* = 0.3\mathcal{X}^{J_1} + 0.2\mathcal{X}^{J_2} + 0.5\mathcal{X}^{J_3}$$

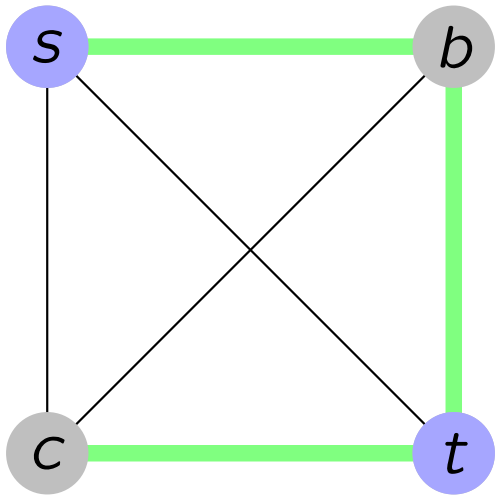
The Randomized Christofides: Made-up Example

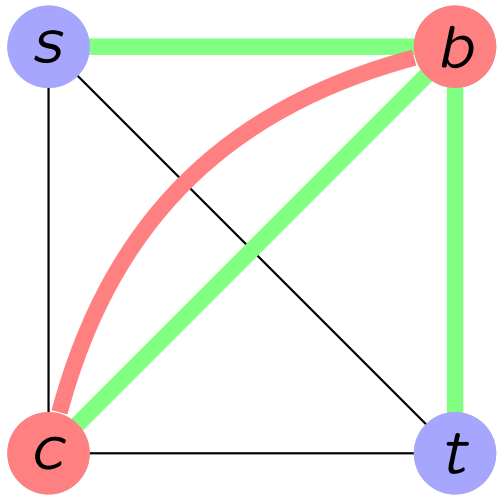
$\mathcal{X}^{J_1}$:
$[1, 0, 1, 0, 1, 0]^T$

$\mathcal{X}^{J_2}$:
$[0, 1, 1, 1, 0, 0]^T$

$\mathcal{X}^{J_3}$:
$[0, 1, 1, 0, 1, 0]^T$







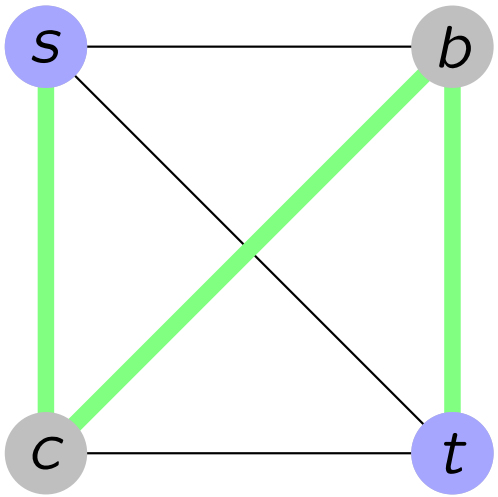
$$\mathbf{x}^* = 0.3\mathcal{X}^{J_1} + 0.2\mathcal{X}^{J_2} + 0.5\mathcal{X}^{J_3}$$

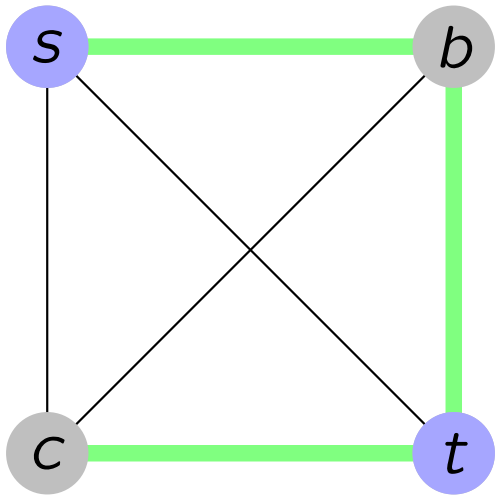
The Randomized Christofides: Made-up Example

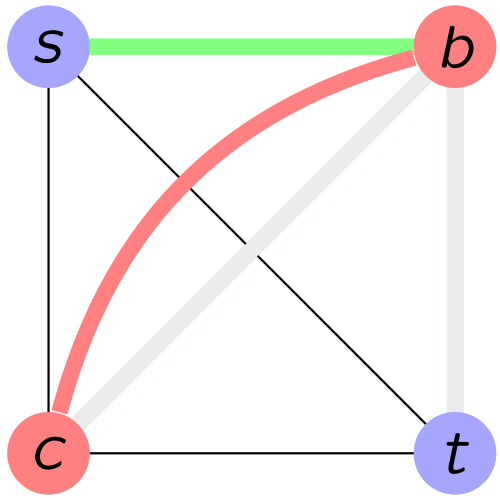
$\mathcal{X}^{J_1}$:
$[1, 0, 1, 0, 1, 0]^T$

$\mathcal{X}^{J_2}$:
$[0, 1, 1, 1, 0, 0]^T$

$\mathcal{X}^{J_3}$:
$[0, 1, 1, 0, 1, 0]^T$







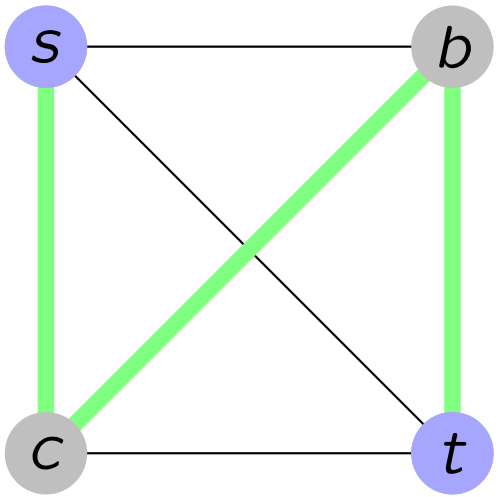
$$\mathbf{x}^* = 0.3\mathcal{X}^{J_1} + 0.2\mathcal{X}^{J_2} + 0.5\mathcal{X}^{J_3}$$

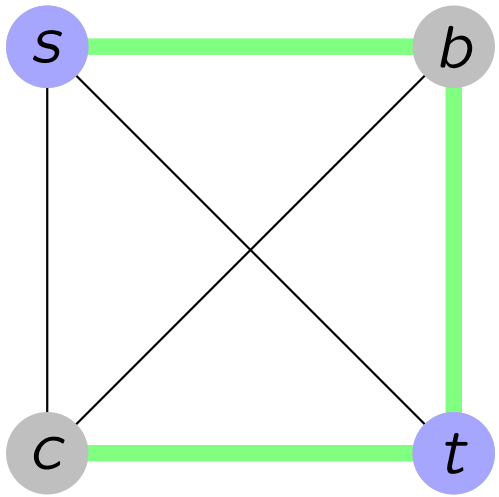
The Randomized Christofides: Made-up Example

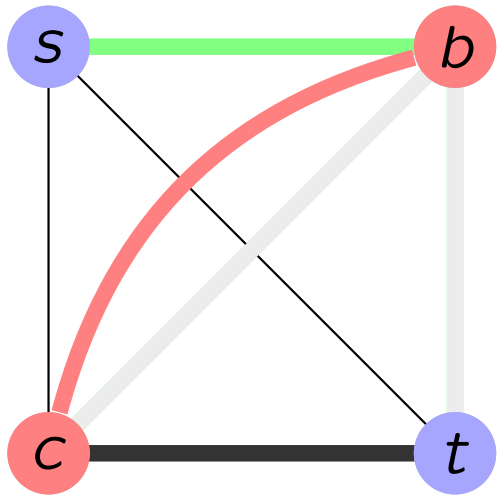
$\mathcal{X}^{J_1}$:
$[1, 0, 1, 0, 1, 0]^T$

$\mathcal{X}^{J_2}$:
$[0, 1, 1, 1, 0, 0]^T$

$\mathcal{X}^{J_3}$:
$[0, 1, 1, 0, 1, 0]^T$







$$\mathbf{x}^* = 0.3\mathcal{X}^{J_1} + 0.2\mathcal{X}^{J_2} + 0.5\mathcal{X}^{J_3}$$

The Randomized Christofides: Analysis

$$\mathbb{E}(\mathbf{c}(\mathbf{HamPath})) \leq \mathbb{E}(\mathbf{c}(\mathbf{J})) + \mathbb{E}(\mathbf{c}(\mathbf{F}))$$

The Randomized Christofides: Analysis

$$\mathbb{E}(\mathbf{c}(\mathbf{HamPath})) \leq \mathbb{E}(\mathbf{c}(\mathbf{J})) + \mathbb{E}(\mathbf{c}(\mathbf{F}))$$

$$\mathbf{x}^* = \sum p_i \boldsymbol{\chi}^{J_i}$$

The Randomized Christofides: Analysis

$$\mathbb{E}(\mathbf{c}(\mathbf{HamPath})) \leq \mathbb{E}(\mathbf{c}(\mathbf{J})) + \mathbb{E}(\mathbf{c}(\mathbf{F}))$$

$$\mathbf{x}^* = \sum p_i \boldsymbol{\chi}^{J_i}$$

$$\mathbb{E}(\mathbf{c}(\mathbf{J})) = \mathbf{c}(\mathbf{x}^*)$$

The Randomized Christofides: Analysis

$$\mathbb{E}(\mathbf{c}(\mathbf{HamPath})) \leq \mathbb{E}(\mathbf{c}(\mathbf{J})) + \mathbb{E}(\mathbf{c}(\mathbf{F}))$$

$$\mathbf{x}^* = \sum p_i \boldsymbol{\chi}^{J_i}$$

$$\mathbb{E}(\mathbf{c}(\mathbf{J})) = \mathbf{c}(\mathbf{x}^*)$$

$$\mathbb{E}(\mathbf{c}(\mathbf{HamPath})) \leq \mathbf{c}(\mathbf{x}^*) + \mathbb{E}(\mathbf{c}(\mathbf{F}))$$

The Randomized Christofides: Analysis

$$\mathbb{E}(\mathbf{c}(\mathbf{HamPath})) \leq \mathbb{E}(\mathbf{c}(\mathbf{J})) + \mathbb{E}(\mathbf{c}(\mathbf{F}))$$

$$\mathbf{x}^* = \sum p_i \boldsymbol{\chi}^{J_i}$$

$$\mathbb{E}(\mathbf{c}(\mathbf{J})) = \mathbf{c}(\mathbf{x}^*)$$

$$\mathbb{E}(\mathbf{c}(\mathbf{HamPath})) \leq \mathbf{c}(\mathbf{x}^*) + \mathbb{E}(\mathbf{c}(\mathbf{F}))$$

$$c(\mathbf{x}^*) \leq c(\mathbf{OPT})$$

Towards bounding $\mathbb{E}(\mathbf{c}(\mathbf{F}))$

- An, Kleinberg, Shmoys (2012):
  - $\mathbb{E}(\mathbf{c}(\mathbf{F})) \leq 0.618c(\mathbf{x}^*) \leq 0.618c(\mathbf{OPT})$
  - $\mathbb{E}(\mathbf{c}(\mathbf{HamPath})) \leq \mathbf{c}(\mathbf{x}^*) + \mathbf{0.618c}(\mathbf{x}^*) \leq \mathbf{1.618c}(\mathbf{OPT})$
  - **1.618**-approximation algorithm

Towards bounding $\mathbb{E}(\mathbf{c}(\mathbf{F}))$

- An, Kleinberg, Shmoys (2012):
  - $\mathbb{E}(\mathbf{c}(\mathbf{F})) \leq 0.618c(\mathbf{x}^*) \leq 0.618c(\mathbf{OPT})$
  - $\mathbb{E}(\mathbf{c}(\mathbf{HamPath})) \leq \mathbf{c}(\mathbf{x}^*) + \mathbf{0.618c}(\mathbf{x}^*) \leq \mathbf{1.618c}(\mathbf{OPT})$
  - **1.618**-approximation algorithm
- Sebo (2013)
  - $\mathbb{E}(\mathbf{c}(\mathbf{F})) \leq 0.6c(\mathbf{x}^*)$.
  - **1.6**-approximation algorithm

Towards bounding $\mathbb{E}(\mathbf{c}(\mathbf{F}))$

- An, Kleinberg, Shmoys (2012):
  - $\mathbb{E}(\mathbf{c}(\mathbf{F})) \leq 0.618c(\mathbf{x}^*) \leq 0.618c(\mathbf{OPT})$
  - $\mathbb{E}(\mathbf{c}(\mathbf{HamPath})) \leq \mathbf{c}(\mathbf{x}^*) + \mathbf{0.618c}(\mathbf{x}^*) \leq \mathbf{1.618c}(\mathbf{OPT})$
  - **1.618**-approximation algorithm
- Sebo (2013)
  - $\mathbb{E}(\mathbf{c}(\mathbf{F})) \leq 0.6c(\mathbf{x}^*)$.
  - **1.6**-approximation algorithm
- Gottschalk, Vygen (2015)
  - Better convex combination.
  - **1.566**-approximation algorithm

Towards bounding $\mathbb{E}(\mathbf{c}(\mathbf{F}))$

- An, Kleinberg, Shmoys (2012):
  - $\mathbb{E}(\mathbf{c}(\mathbf{F})) \leq 0.618c(\mathbf{x}^*) \leq 0.618c(\mathbf{OPT})$
  - $\mathbb{E}(\mathbf{c}(\mathbf{HamPath})) \leq \mathbf{c}(\mathbf{x}^*) + \mathbf{0.618c}(\mathbf{x}^*) \leq \mathbf{1.618c}(\mathbf{OPT})$
  - **1.618**-approximation algorithm
- Sebo (2013)
  - $\mathbb{E}(\mathbf{c}(\mathbf{F})) \leq 0.6c(\mathbf{x}^*)$.
  - **1.6**-approximation algorithm
- Gottschalk, Vygen (2015)
  - Better convex combination.
  - **1.566**-approximation algorithm
- Sebo (2016)
  - **1.53**-approximation algorithm

Towards bounding $\mathbb{E}(\mathbf{c}(\mathbf{F}))$

- An, Kleinberg, Shmoys (2012):
  - $\mathbb{E}(\mathbf{c}(\mathbf{F})) \leq 0.618c(\mathbf{x}^*) \leq 0.618c(\mathbf{OPT})$
  - $\mathbb{E}(\mathbf{c}(\mathbf{HamPath})) \leq \mathbf{c}(\mathbf{x}^*) + \mathbf{0.618c}(\mathbf{x}^*) \leq \mathbf{1.618c}(\mathbf{OPT})$
  - **1.618**-approximation algorithm
- Sebo (2013)
  - $\mathbb{E}(\mathbf{c}(\mathbf{F})) \leq 0.6c(\mathbf{x}^*)$.
  - **1.6**-approximation algorithm
- Gottschalk, Vygen (2015)
  - Better convex combination.
  - **1.566**-approximation algorithm
- Sebo (2016)
  - **1.53**-approximation algorithm
- LP-based **1.5**-approximation algorithm?

Thank you! Questions?