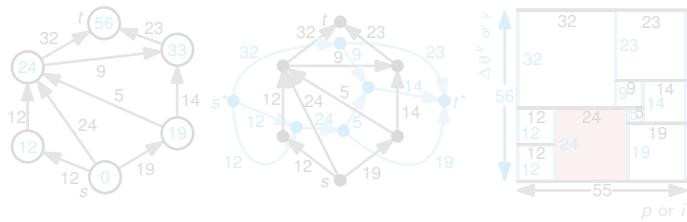


Problem Statements

Feasible electrical flows



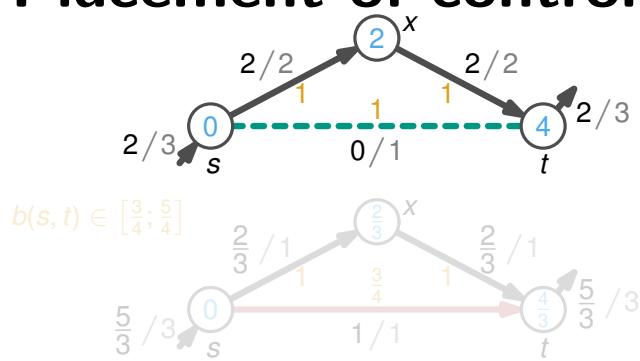
Dynamic models

Static models

Alternating Current (AC) model

Direct Current (DC) model

Placement of control units



Discrete control decision

[Grastien, Rutter, Wagner, W., and Wolf, 2018]

Continuous control decision

[Leibfried, Mchedlidze, Meyer-Hübner, Nöllenburg, Rutter, Sanders, Wagner, and W., 2015]

[Mchedlidze, Nöllenburg, Rutter, Wagner, and W., 2015]

Cable layout



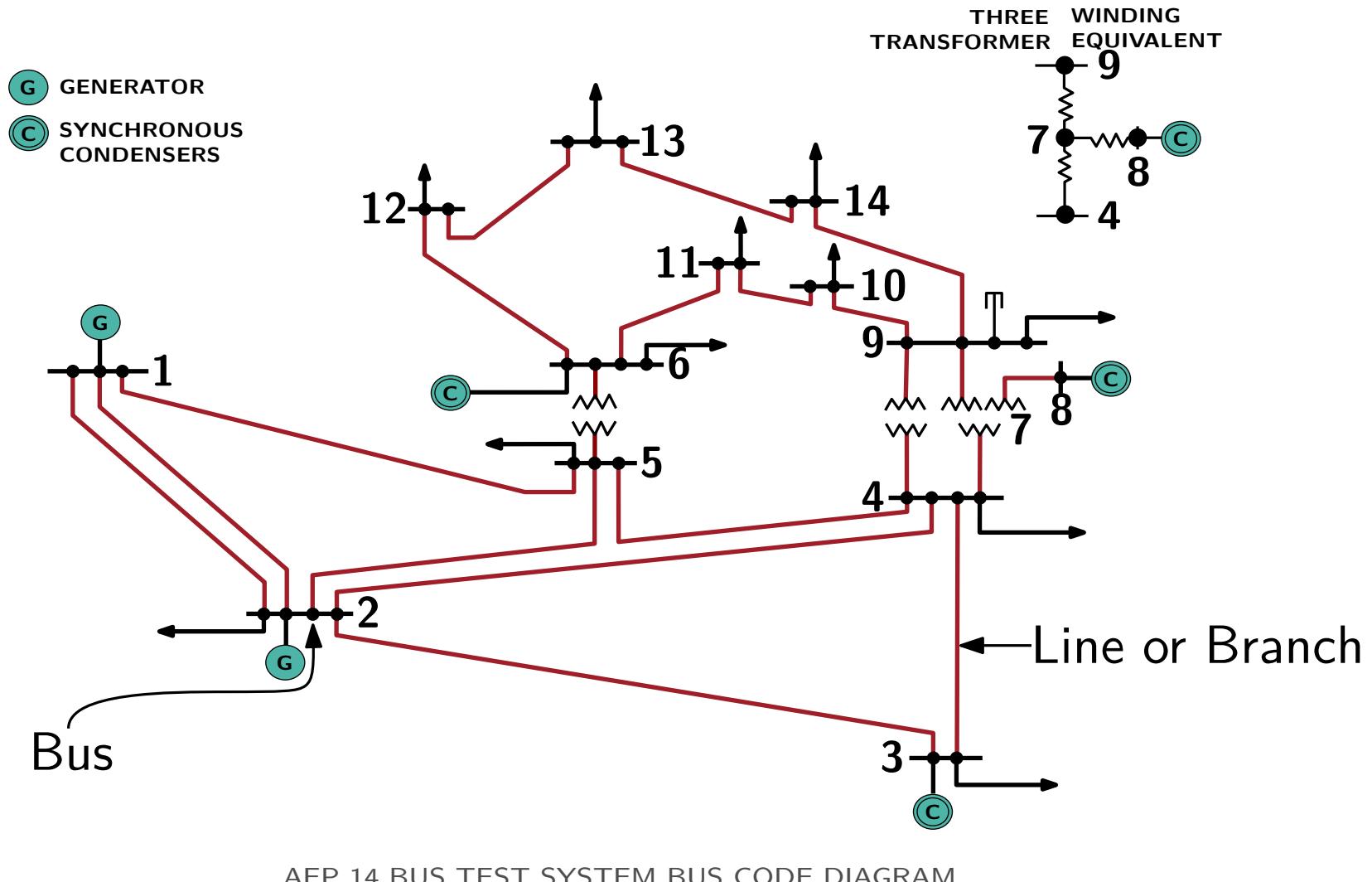
Wind farm cabling with multiple cable types

[Lehmann, Rutter, Wagner, and W., 2017]

From a Transmission Network to a Graph

[University of Washington, 1999]

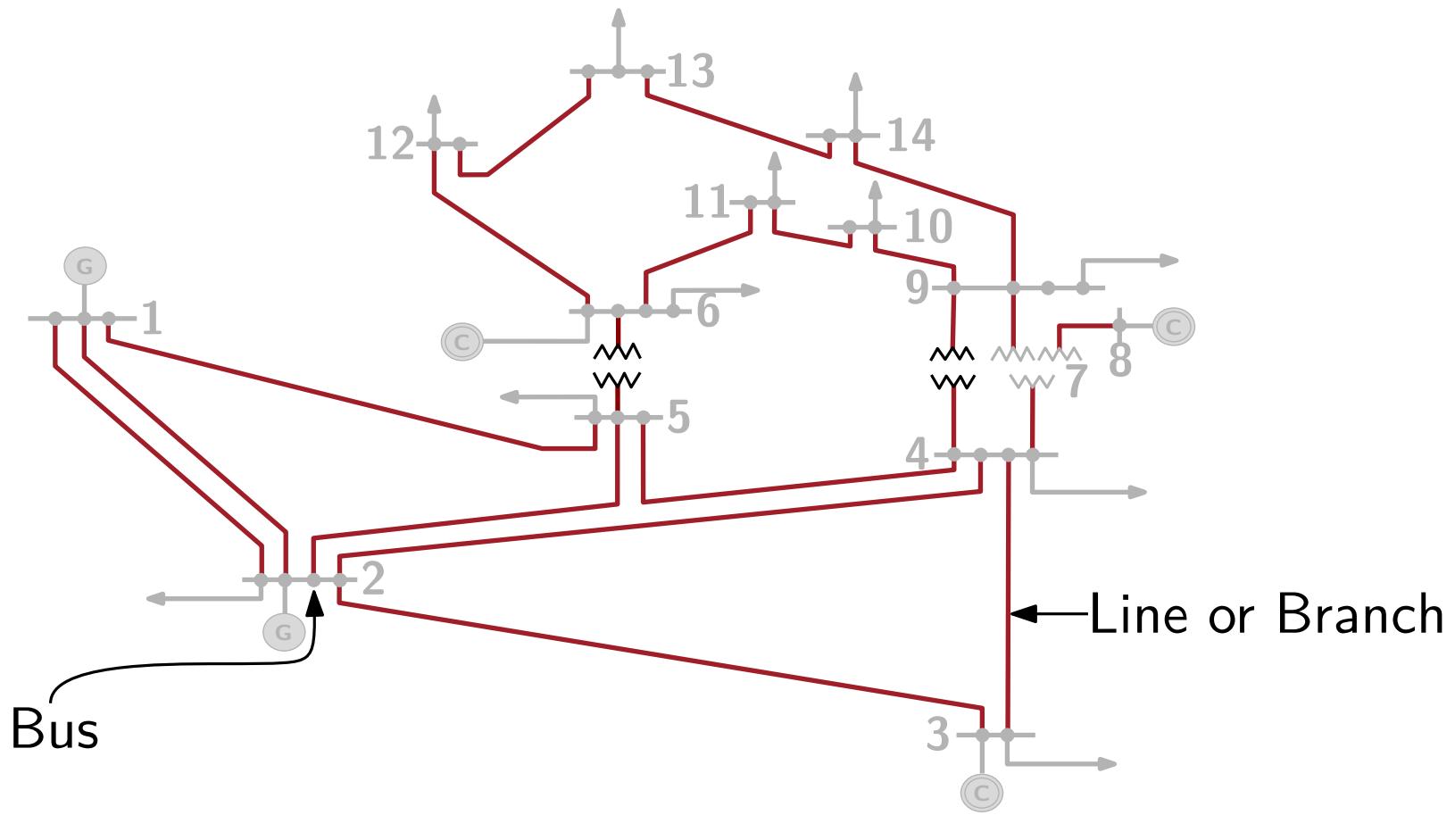
Graph $G = (V, E)$



From a Transmission Network to a Graph

[University of Washington, 1999]

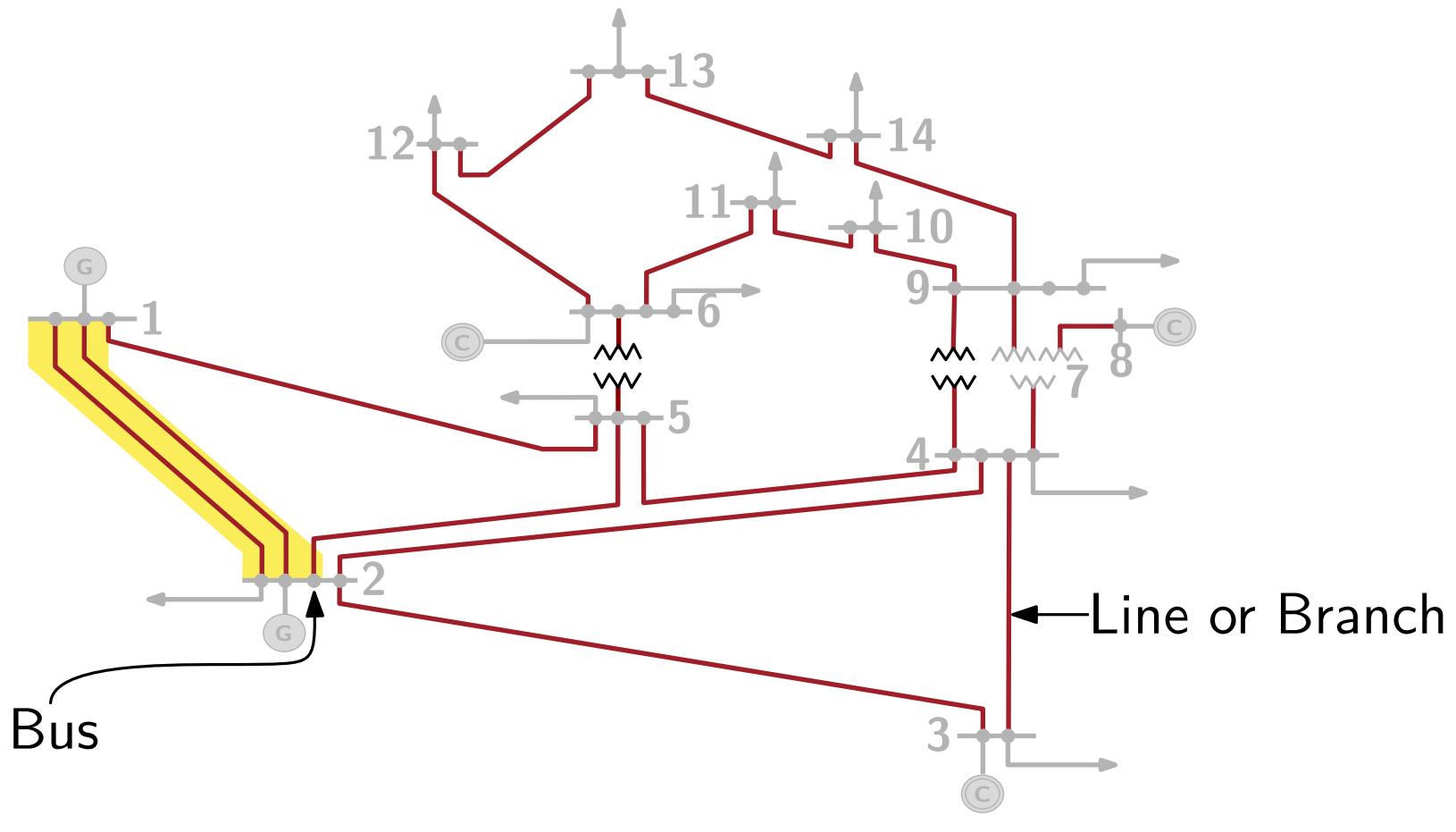
Graph $G = (V, E)$



From a Transmission Network to a Graph

[University of Washington, 1999]

Graph $G = (V, E)$

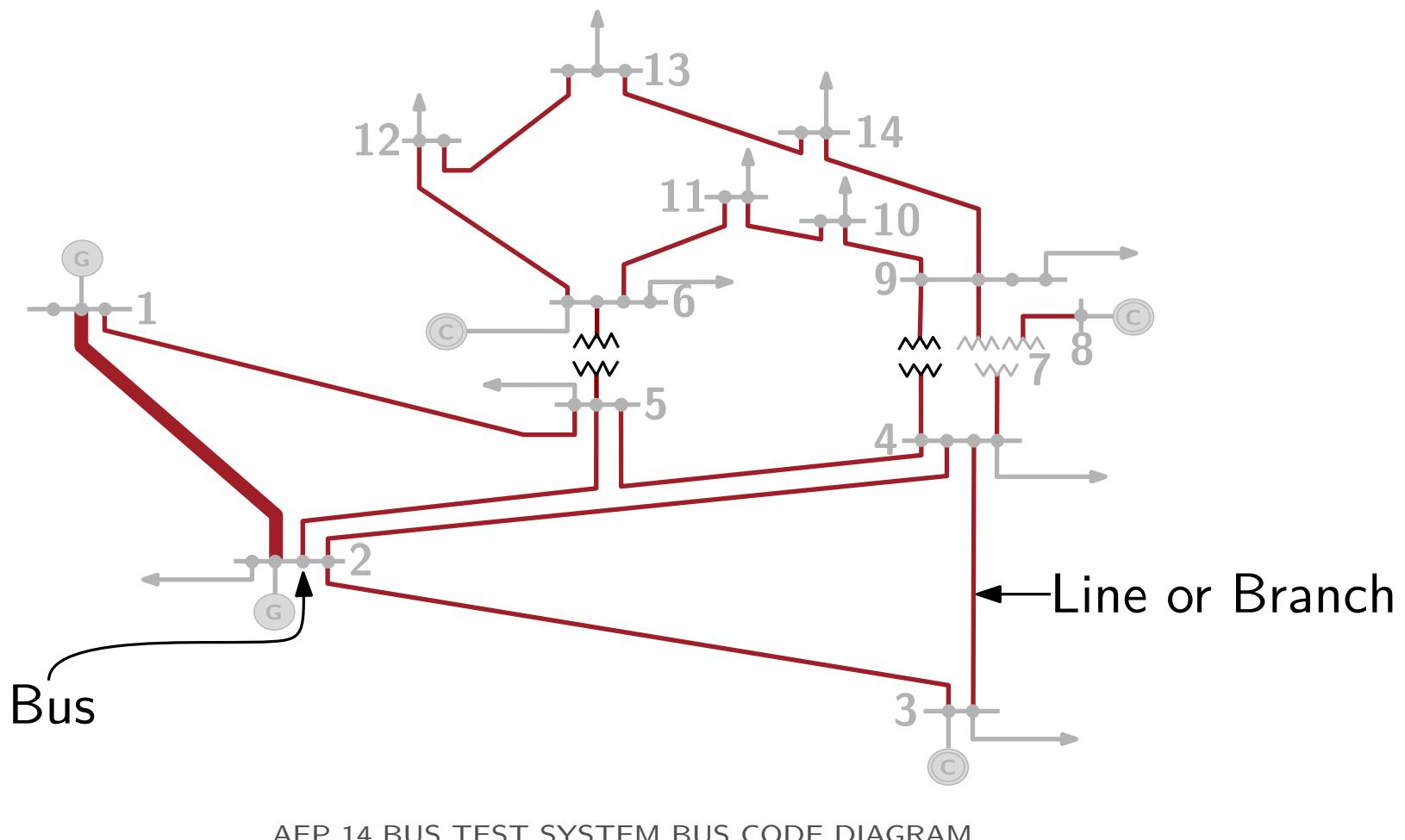


AEP 14 BUS TEST SYSTEM BUS CODE DIAGRAM

From a Transmission Network to a Graph

[University of Washington, 1999]

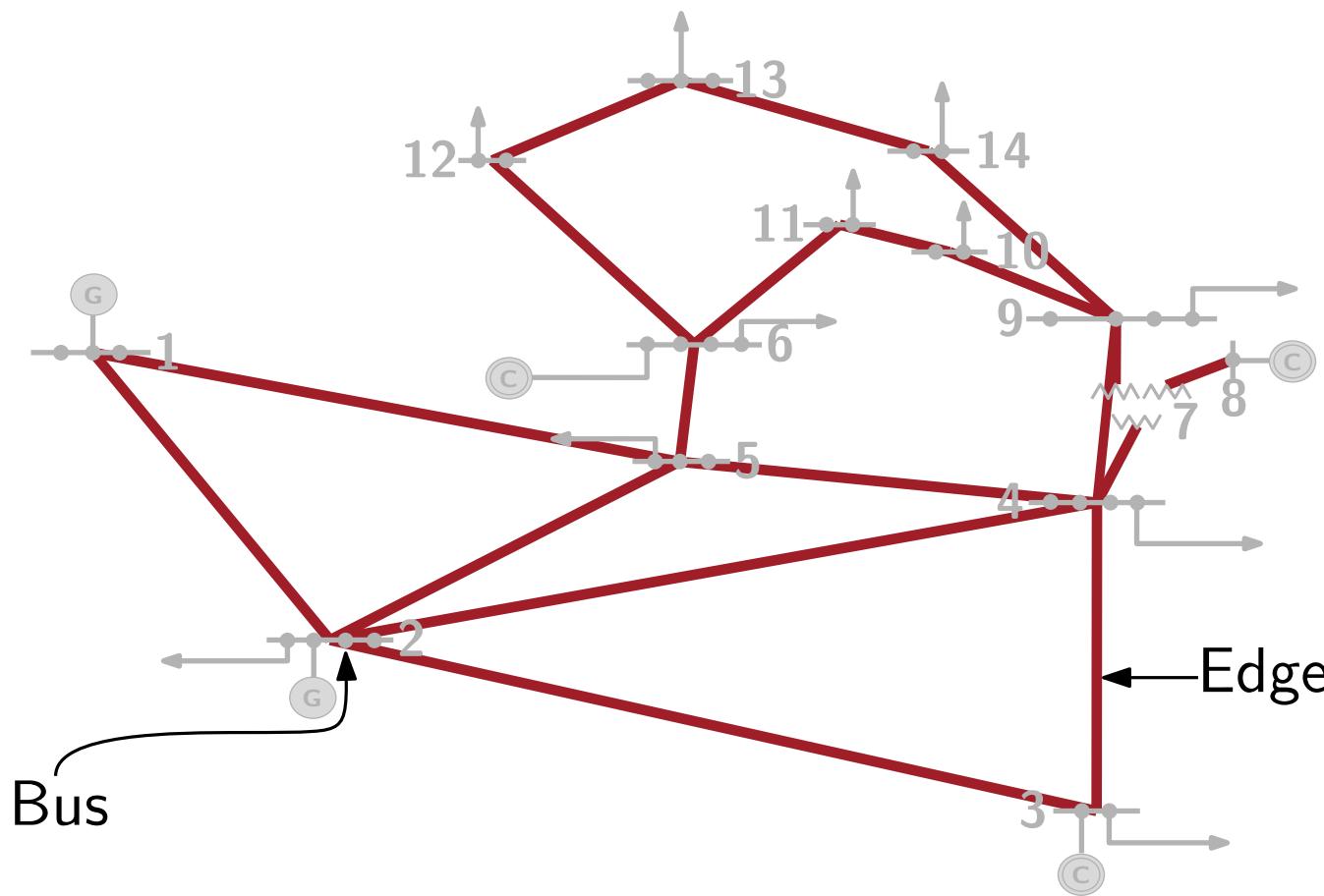
Graph $G = (V, E)$



From a Transmission Network to a Graph

[University of Washington, 1999]

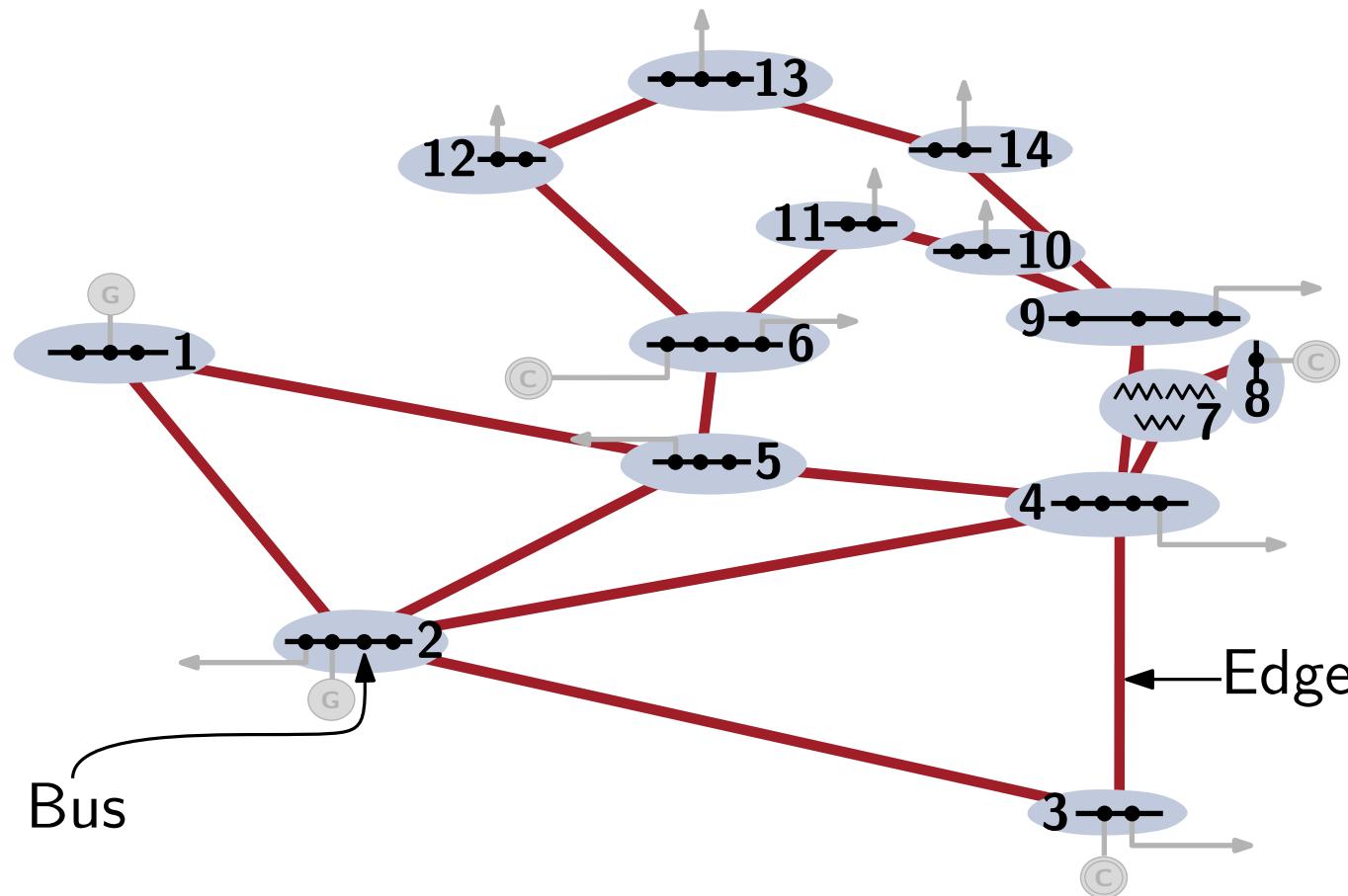
Graph $G = (V, E)$



From a Transmission Network to a Graph

[University of Washington, 1999]

Graph $G = (V, E)$

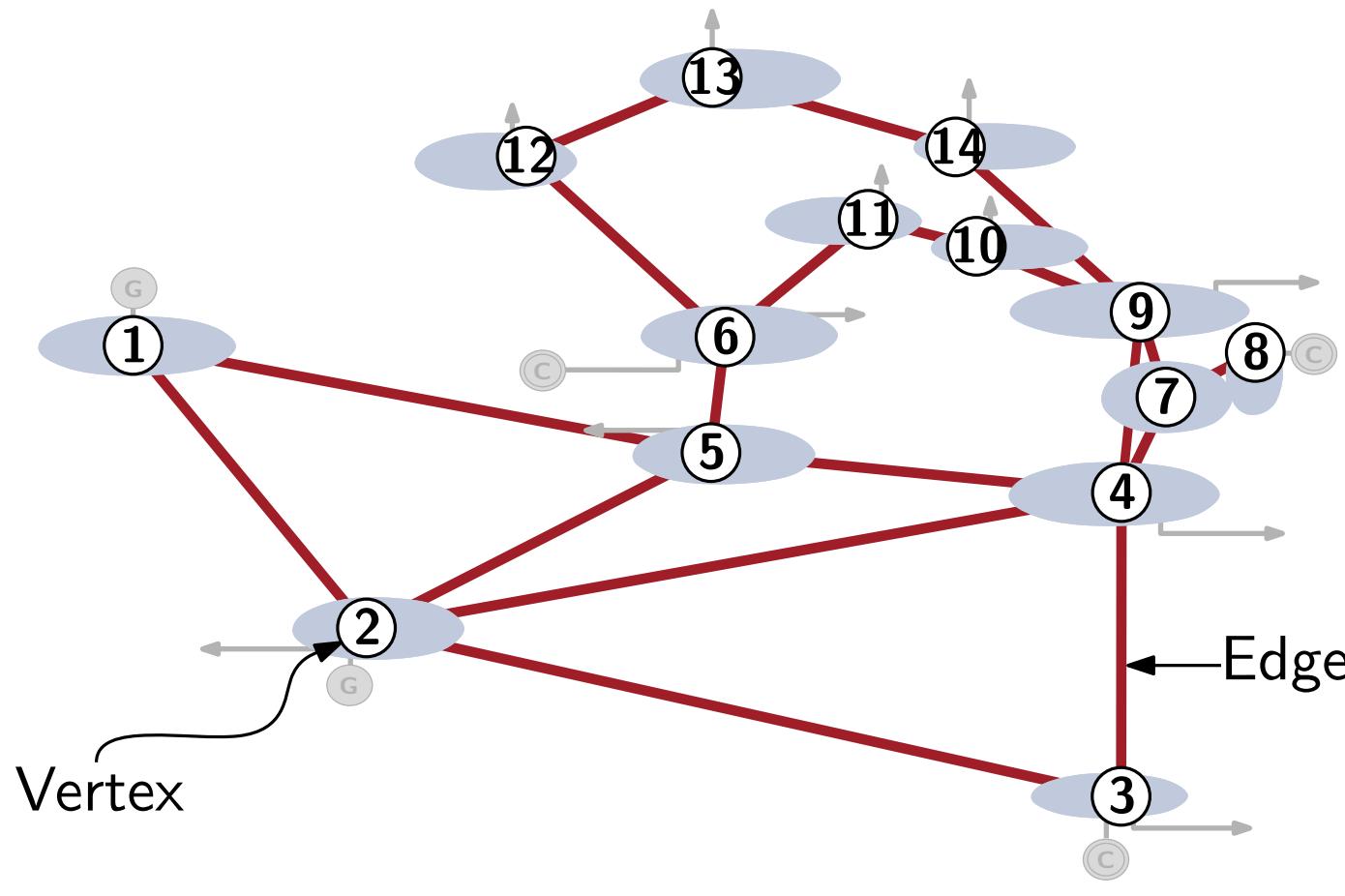


AEP 14 BUS TEST SYSTEM BUS CODE DIAGRAM

From a Transmission Network to a Graph

[University of Washington, 1999]

Graph $G = (V, E)$

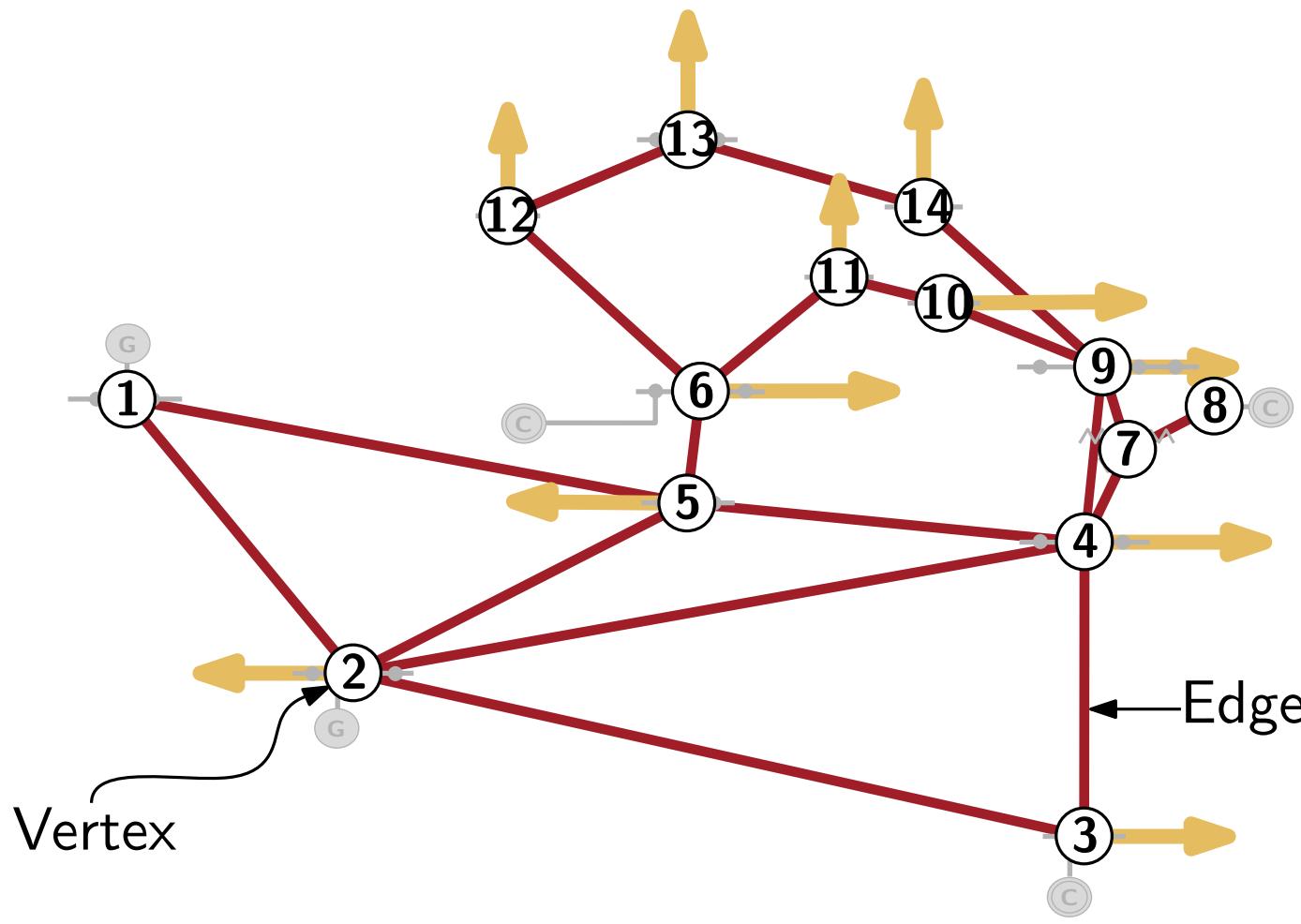


AEP 14 BUS TEST SYSTEM BUS CODE DIAGRAM

From a Transmission Network to a Graph

[University of Washington, 1999]

Graph $G = (V, E)$

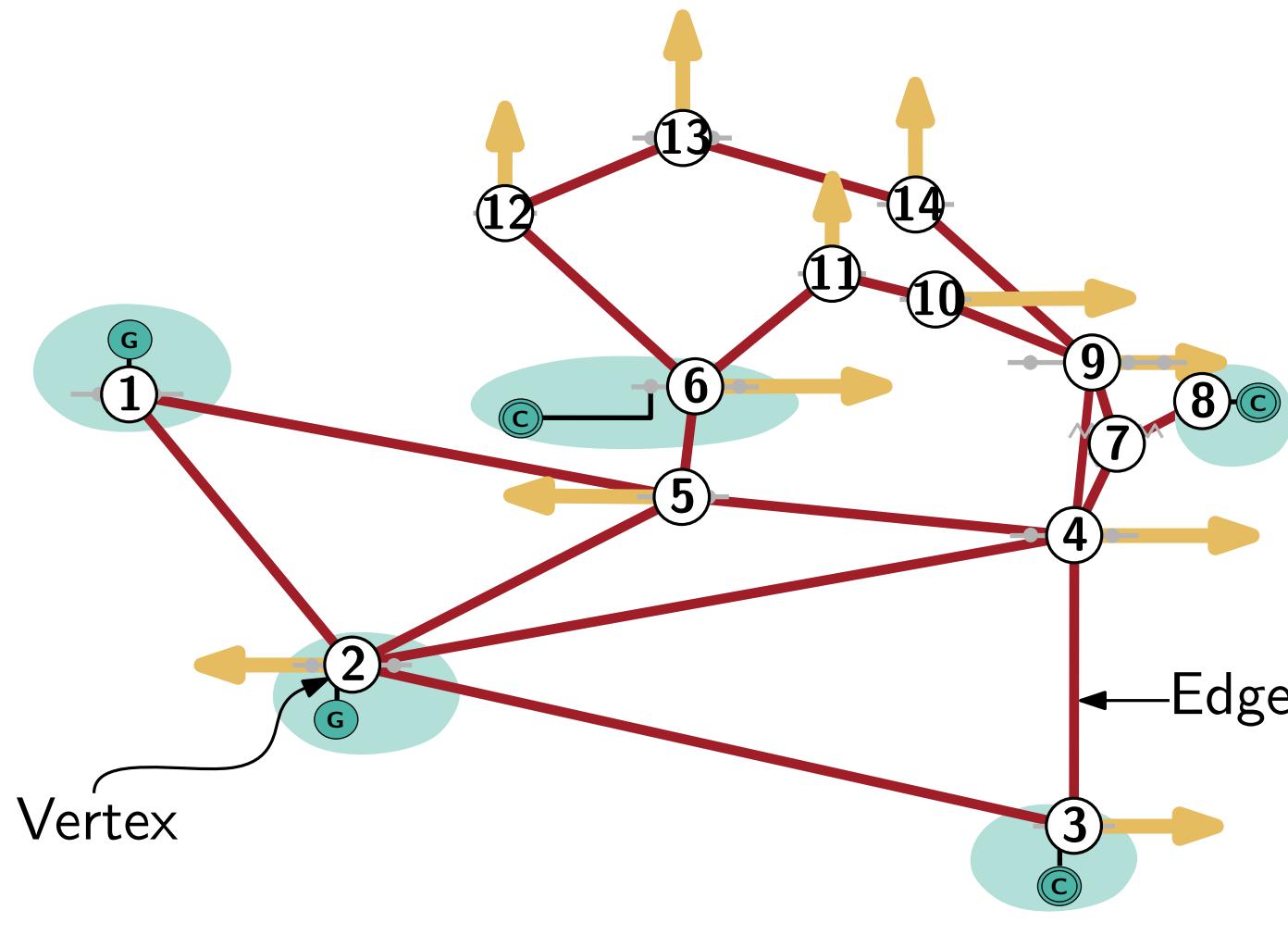


AEP 14 BUS TEST SYSTEM BUS CODE DIAGRAM

From a Transmission Network to a Graph

[University of Washington, 1999]

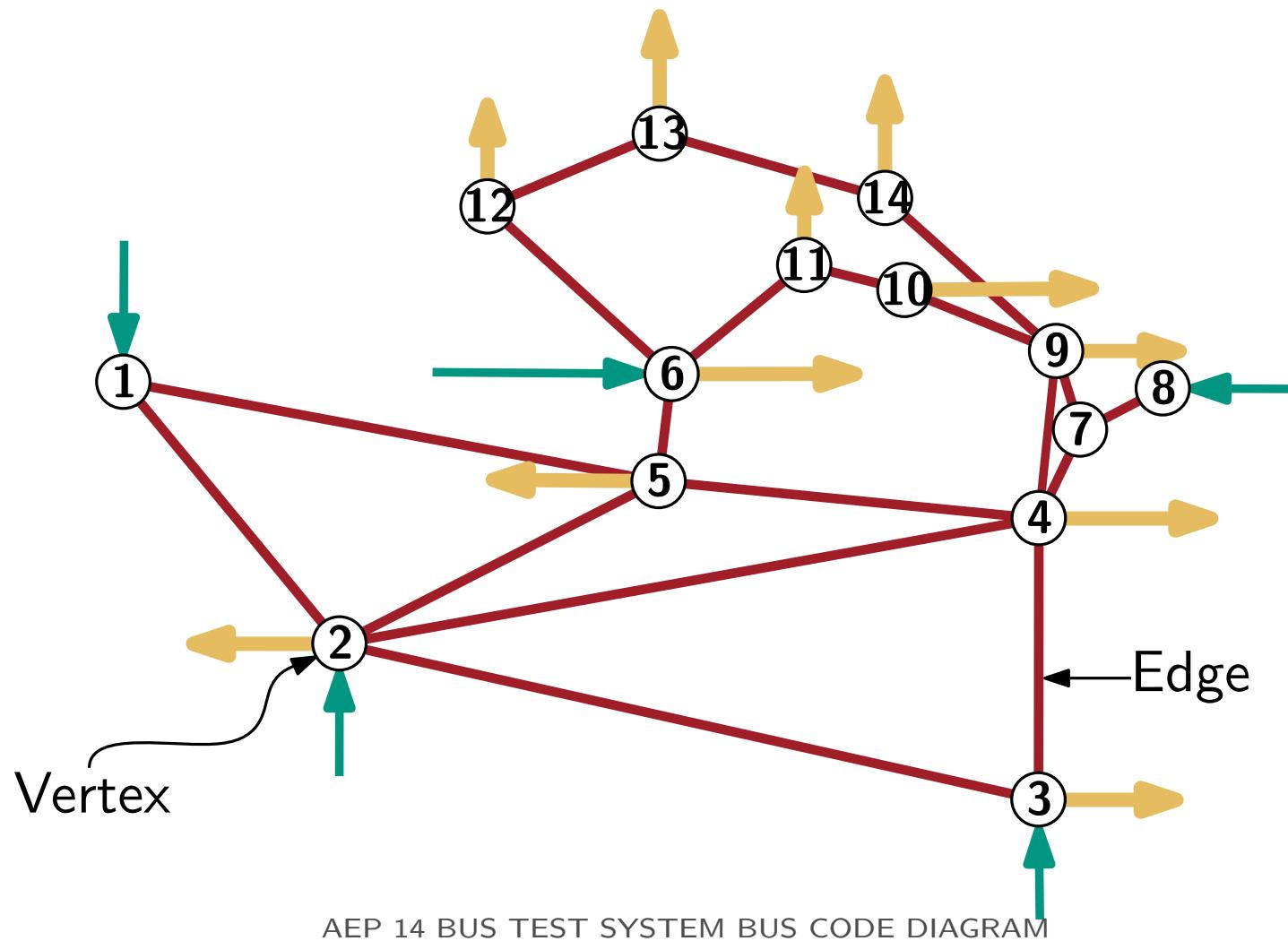
Graph $G = (V, E)$



From a Transmission Network to a Graph

[University of Washington, 1999]

Graph $G = (V, E)$



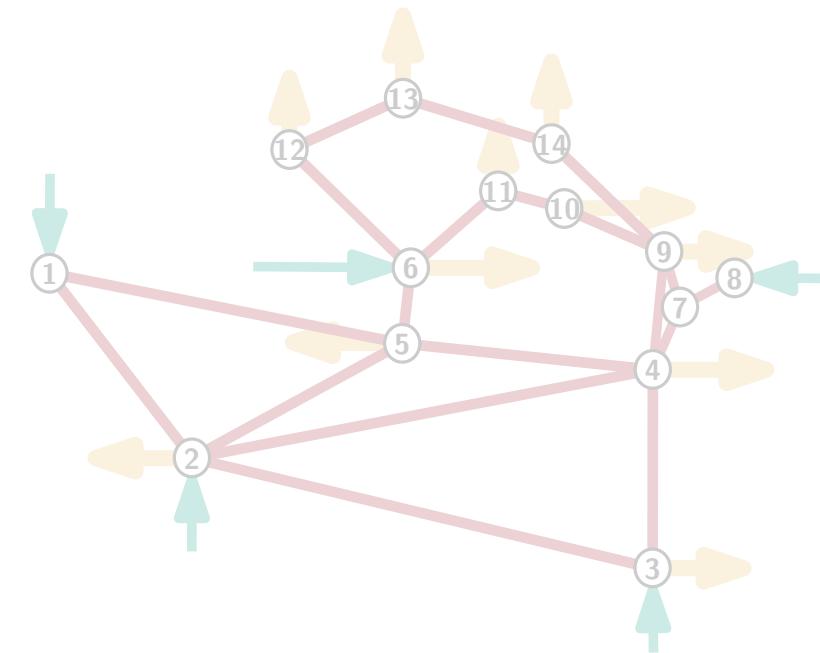
Input

- Let $G = (V, E)$ be the bidirected graph and $\underline{G} = (V, \overleftrightarrow{E})$ the underlying undirected graph.
- Set of vertices V (also called buses) with generators $V_G \subseteq V$, consumers $V_D \subseteq V \setminus V_G$, and intermediate vertices $V \setminus (V_G \cup V_D)$
- We denote by \overleftrightarrow{E} the underlying undirected edge set with $\overleftrightarrow{e} \in \overleftrightarrow{E}$ such that $(u, v) = \underline{(v, u)}$
- Network $\mathcal{N} = (G, V_G, V_D, \text{cap}, \underline{b}, \underline{d})$
thermal line limits $\text{cap}: E \rightarrow \mathbb{R}_{\geq 0}$,
susceptance $b: E \rightarrow \mathbb{R}_{\geq 0}$,
demands lower bound $\underline{d}: V_D \rightarrow \mathbb{R}_{\geq 0}$.

The Maximum Transmission Switching Flow Problem

Given V set of buses, $V_L \subseteq V$ set of loads (with capacities),
 $V_G \subseteq V$ set of generators (with capacities)
 E set of lines (each with impedance, susceptance, capacity)

inputs



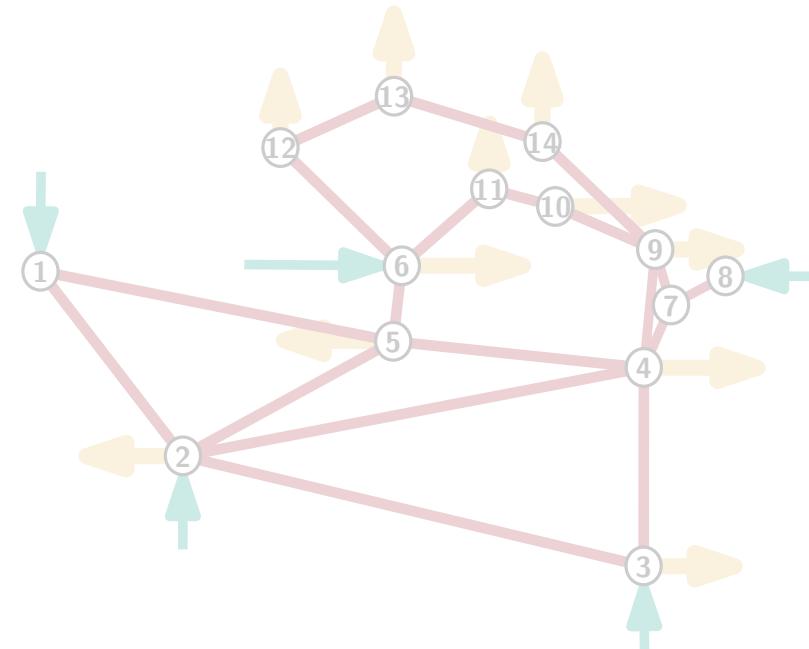
The Maximum Transmission Switching Flow Problem

Given V set of buses, $V_L \subseteq V$ set of loads (with capacities),

$V_G \subseteq V$ set of generators (with capacities)

E set of lines (each with impedance, susceptance, capacity)

find for each line: **if the line is switched** variables



The Maximum Transmission Switching Flow Problem

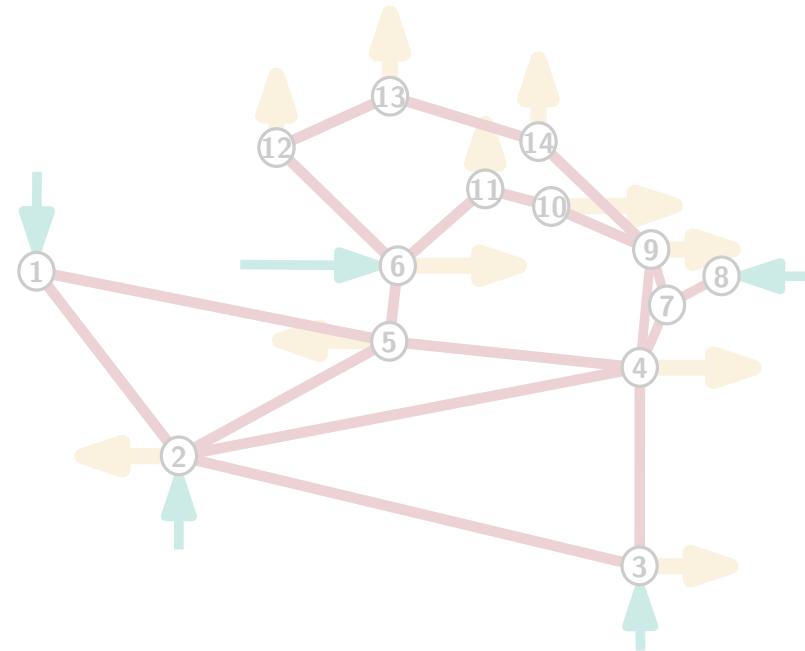
Given V set of buses, $V_L \subseteq V$ set of loads (with capacities),

$V_G \subseteq V$ set of generators (with capacities)

E set of lines (each with impedance, susceptance, capacity)

find for each line: if the line is switched

maximize power production objective



The Maximum Transmission Switching Flow Problem

Given V set of buses, $V_L \subseteq V$ set of loads (with capacities),

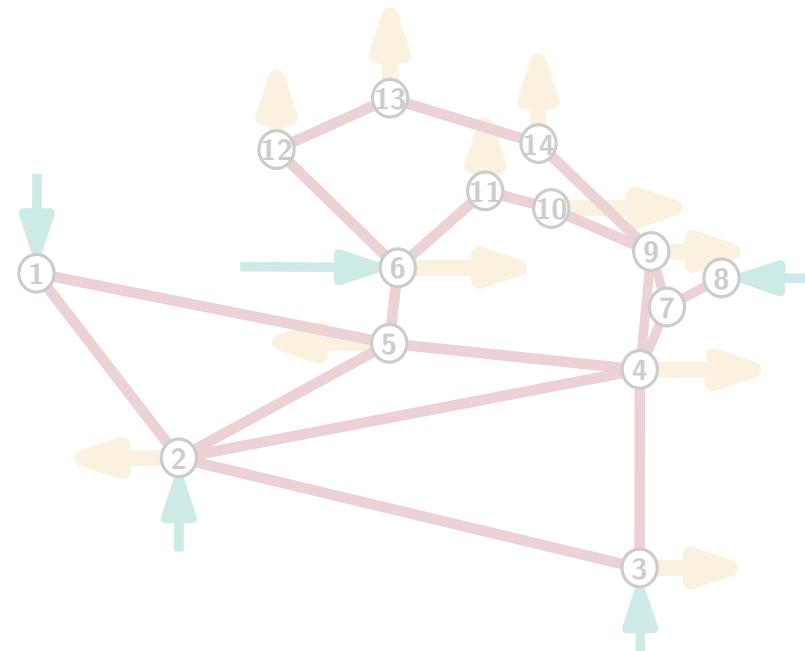
$V_G \subseteq V$ set of generators (with capacities)

E set of lines (each with impedance, susceptance, capacity)

find for each line: if the line is switched

maximize power production

subject to line capacity constraints
load capacity constraints
power flow constraints



The Maximum Transmission Switching Flow Problem

Given V set of buses, $V_L \subseteq V$ set of loads (with capacities),

$V_G \subseteq V$ set of generators (with capacities)

E set of lines (each with impedance, susceptance, capacity)

The **AC** conservation of flow is a **subproblem** of the MTSF problem.

AC conservation of flow is already **NP-hard on trees**.

[Lehmann et al., 2015]

subject to
line capacity constraints
load capacity constraints
power flow constraints



The Maximum Transmission Switching Flow Problem

Given V set of buses, $V_L \subseteq V$ set of loads (with capacities),

$V_G \subseteq V$ set of generators (with capacities)

E set of lines (each with impedance, susceptance, capacity)

The **AC** conservation of flow is a **subproblem** of the MTSF problem.

AC conservation of flow is already **NP-hard on trees**.

[Lehmann et al., 2015]

→ Power grids are not easy.

→ **Linearized AC** conservation of flow is **easy** to solve.

load capacity constraints

power flow constraints



Feasible Flow

- A flow is a function $f: E \rightarrow \mathbb{R}$ with skew-symmetry $f(u, v) = -f(v, u)$ for all $(u, v) \in E$.
- The net flow $f_{\text{net}}(u) := \sum_{\{u,v\} \in \vec{E}} f(u, v)$
- Flow f satisfies the following conservation of flow properties that are similar to **Kirchhoff's Current Law (KCL)**

$$f_{\text{net}}(u) = 0 \quad \forall u \in V \setminus (V_G \cup V_D)$$

$$-\infty \leq f_{\text{net}}(u) \leq -d \quad \forall u \in V_D$$

$$0 \leq f_{\text{net}}(u) \leq \infty \quad \forall u \in V_G$$

- Flow f is called *feasible* if

$$|f(u, v)| \leq \text{cap}(u, v) \quad \forall (u, v) \in E$$

- Flow value $F(\mathcal{N}, f)$ of flow f on \mathcal{N} is defined by

$$\sum_{u \in V_G} f_{\text{net}}(u)$$

The maximum flow (MF) Problem

- Flow value $F(\mathcal{N}, \mathbf{f})$ of flow \mathbf{f} on \mathcal{N} is defined by

$$\sum_{u \in V_G} f_{\text{net}}(u)$$

- The maximum flow (MF) is denoted by $\text{MF}(\mathcal{N})$

$$\text{OPT}_{\text{MF}}(\mathcal{N}) = \max F(\mathcal{N}, \mathbf{f})$$

with \mathbf{f} being a feasible flow meaning

$$f_{\text{net}}(u) = 0 \quad \forall u \in V \setminus (V_G \cup V_D)$$

$$-\infty \leq f_{\text{net}}(u) \leq -d \quad \forall u \in V_D$$

$$0 \leq f_{\text{net}}(u) \leq \infty \quad \forall u \in V_G$$

$$|f(u, v)| \leq \text{cap}(u, v) \quad \forall (u, v) \in E$$

Physical Feasible Flow

- A **feasible** flow neglects **physical circumstances**
- The Kirchhoff's Voltage Law (KVL) is one of them with potentials at each vertex $\theta: V \rightarrow \mathbb{R}$

$$b(u, v) \cdot (\theta(v) - \theta(u) - \theta_{\text{shift}}) = f(u, v) \quad \forall (u, v) \in E$$
$$\theta_{\min}(u) \leq \theta(u) \leq \theta_{\max}(u) \quad \forall u \in V$$

- Note that $\theta_{\text{shift}}(u, v) = 0$ if there is no transformer that is assumed in general and thus, is neglected

The maximum power flow (MPF) Problem

- The maximum power flow (MPF) is denoted by $\text{MPF}(\mathcal{N})$ with value

$$\text{OPT}_{\text{MPF}}(\mathcal{N}) = \max F(\mathcal{N}, f)$$

with f being a *physical feasible* flow meaning

$$f_{\text{net}}(u) = 0 \quad \forall u \in V \setminus (V_G \cup V_D)$$

$$-\infty \leq f_{\text{net}}(u) \leq -d_u \quad \forall u \in V_D$$

$$0 \leq f_{\text{net}}(u) \leq \infty \quad \forall u \in V_G$$

$$|f(u, v)| \leq \text{cap}(u, v) \quad \forall (u, v) \in E$$

$$b(u, v) \cdot (\theta(v) - \theta(u) - \theta_{\text{shift}}) = f(u, v) \quad \forall (u, v) \in E$$

$$\theta_{\min}(u) \leq \theta(u) \leq \theta_{\max}(u) \quad \forall u \in V$$

The maximum power flow (MPF) Problem

- The maximum power flow (MPF) is denoted by $\text{MPF}(\mathcal{N})$ with value

$$\text{OPT}_{\text{MPF}}(\mathcal{N}) = \max F(\mathcal{N}, f)$$

with f being a *physical feasible* flow meaning

$$f_{\text{net}}(u) = 0 \quad \forall u \in V \setminus (V_G \cup V_D)$$

$$-\infty \leq f_{\text{net}}(u) \leq -d_u \quad \forall u \in V_D$$

$$0 \leq f_{\text{net}}(u) \leq \infty \quad \forall u \in V_G$$

$$|f(u, v)| \leq \text{cap}(u, v) \quad \forall (u, v) \in E$$

$$b(u, v) \cdot (\theta(v) - \theta(u)) = f(u, v) \quad \forall (u, v) \in E$$

$$\theta_{\min}(u) \leq \theta(u) \leq \theta_{\max}(u) \quad \forall u \in V$$

The Maximum Transmission Switching Flow (MTSF) Problem

[Lehmann et al., 2014]

- The MAXIMUM TRANSMISSION SWITCHING FLOW (MTSF) is denoted by

$$\text{MTSF}(\mathcal{N}) := \max_{S \subseteq \overleftarrow{E}} \text{MPF}(\mathcal{N} - S)$$

with value $\text{OPT}_{\text{MTSF}}(\mathcal{N})$ with f being a **physical feasible** flow meaning

$$\begin{aligned} f_{\text{net}}(u) &= 0 & \forall u \in V \setminus (V_G \cup V_D) \\ -\infty \leq f_{\text{net}}(u) &\leq -d_u & \forall u \in V_D \\ 0 \leq f_{\text{net}}(u) &\leq \infty & \forall u \in V_G \\ |f(u, v)| &\leq z(u, v) \cdot \text{cap}(u, v) & \forall (u, v) \in E \\ b(u, v) \cdot z(u, v) \cdot (\theta(v) - \theta(u)) &= f(u, v) & \forall (u, v) \in E \\ \theta_{\min}(u) \leq \theta(u) &\leq \theta_{\max}(u) & \forall u \in V \end{aligned}$$

The Maximum Transmission Switching Flow (MTSF) Problem

[Lehmann et al., 2014]

- The MAXIMUM TRANSMISSION SWITCHING FLOW (MTSF) is denoted by

$$\text{MTSF}(\mathcal{N}) := \max_{S \subseteq \overleftarrow{E}} \text{MPF}(\mathcal{N} - S)$$

with value $\text{OPT}_{\text{MTSF}}(\mathcal{N})$ with f being a **physical feasible** flow meaning

$$f_{\text{net}}(u) = 0 \quad \forall u \in V \setminus (V_G \cup V_D)$$

$$-\infty \leq f_{\text{net}}(u) \leq -d_u \quad \forall u \in V_D$$

$$0 \leq f_{\text{net}}(u) \leq \infty \quad \forall u \in V_G$$

$$|f(u, v)| \leq z(u, v) \cdot \text{cap}(u, v) \quad \forall (u, v) \in E$$

$$b(u, v) \cdot (\theta(v) - \theta(u)) + (1 - z(u, v))M \geq f(u, v) \quad \forall (u, v) \in E$$

$$b(u, v) \cdot (\theta(v) - \theta(u)) - (1 - z(u, v))M \leq f(u, v) \quad \forall (u, v) \in E$$

$$\theta_{\min}(u) \leq \theta(u) \leq \theta_{\max}(u) \quad \forall u \in V$$

The Maximum Transmission Switching Flow (MTSF) Problem

[Lehmann et al., 2014]

Optimization Problem MTSF

Instance: A power grid \mathcal{N} .

Objective: Find a set $S \subseteq E$ of switched edges such that $\text{OPT}_{\text{MPF}}(\mathcal{N} - S)$ is maximum among all choices of switched edges S .

Decision Problem k -MTSF

Instance: A power grid \mathcal{N} and $k \in \mathbb{Q}_{\geq 0}$.

Objective: Is it possible to remove a set of edges $S \subseteq E$ such that there is an physical **feasible** flow f in $\mathcal{N} - S$ with flow value $F(\mathcal{N} - S, f) \geq k$?

The Optimal Power Flow (OPF) Problem

[Zimmerman et al.,
2011]

- Network $\mathcal{N} = (G, V_G, V_D, \text{cap}, \mathbf{b}, \underline{\mathbf{d}})$
- The OPTIMAL POWER FLOW (OPF) is denoted by $\text{OPF}(\mathcal{N})$ with value

$$\text{OPT}_{\text{OPF}}(\mathcal{N}) := \min \sum_{u \in V_G} \gamma_u(f_{\text{net}}(u))$$

with f being a physical feasible flow meaning
 $f_{\text{net}}(u) = 0$

$$\begin{aligned} -\infty &\leq f_{\text{net}}(u) \leq -\underline{d}_u & \forall u \in V \setminus (V_G \cup V_D) \\ 0 &\leq f_{\text{net}}(u) \leq \infty & \forall u \in V_D \\ |f(u, v)| &\leq \text{cap}(u, v) & \forall u \in V_G \\ b(u, v) \cdot (\theta(v) - \theta(u)) &= f(u, v) & \forall (u, v) \in E \\ \theta_{\min}(u) &\leq \theta(u) \leq \theta_{\max}(u) & \forall (u, v) \in E \\ && \forall u \in V \end{aligned}$$

The Optimal Power Flow (OPF) Problem

[Zimmerman et al.,
2011] $\in \mathbb{R}$

- Network $\mathcal{N}_{\text{bounded}} = (G, V_G, V_D, \text{cap}, \underline{b}, \underline{x}, \bar{x}, \underline{d}, \bar{d})$ is called bounded
- Generation cost function $\gamma_u: \mathbb{R} \rightarrow \mathbb{R}_{\geq 0}$ for all $u \in V_G$
- The OPTIMAL POWER FLOW (OPF) is denoted by $\text{OPF}(\mathcal{N})$ with value

$$\text{OPT}_{\text{OPF}}(\mathcal{N}) := \min \sum_{u \in V_G} \gamma_u(f_{\text{net}}(u))$$

with f being a **physical feasible** flow meaning

$$f_{\text{net}}(u) = 0 \quad \forall u \in V \setminus (V_G \cup V_D)$$

$$-\bar{d}_u \leq f_{\text{net}}(u) \leq -\underline{d}_u \quad \forall u \in V_D$$

$$\underline{x} \leq f_{\text{net}}(u) \leq \bar{x} \quad \forall u \in V_G$$

$$|f(u, v)| \leq \text{cap}(u, v) \quad \forall (u, v) \in E$$

$$b(u, v) \cdot (\theta(v) - \theta(u)) = f(u, v) \quad \forall (u, v) \in E$$

$$\theta_{\min}(u) \leq \theta(u) \leq \theta_{\max}(u) \quad \forall u \in V$$

The Optimal Power Flow (OPF) Problem

[Zimmerman et al.,
2011] $\in \mathbb{R}$

- Network $\mathcal{N}_{\text{bounded}} = (G, V_G, V_D, \text{cap}, \underline{b}, \underline{x}, \bar{x}, \underline{d}, \bar{d})$ is called bounded
- Generation cost function $\gamma_u: \mathbb{R} \rightarrow \mathbb{R}_{\geq 0}$ for all $u \in V_G$
- The OPTIMAL POWER FLOW (OPF) is denoted by $\text{OPF}(\mathcal{N})$ with value

$$\text{OPT}_{\text{OPF}}(\mathcal{N}) := \min \sum_{u \in V_G} \gamma_u(f_{\text{net}}(u))$$

with f being a **physical feasible** flow meaning

$$f_{\text{net}}(u) = 0 \quad \forall u \in V \setminus (V_G \cup V_D)$$

$$-\bar{d}_u = -\bar{d}_u \leq f_{\text{net}}(u) \leq -\underline{d}_u = -d_u \quad \forall u \in V_D$$

$$\underline{x} \leq f_{\text{net}}(u) \leq \bar{x} \quad \forall u \in V_G$$

$$|f(u, v)| \leq \text{cap}(u, v) \quad \forall (u, v) \in E$$

$$b(u, v) \cdot (\theta(v) - \theta(u)) = f(u, v) \quad \forall (u, v) \in E$$

$$\theta_{\min}(u) \leq \theta(u) \leq \theta_{\max}(u) \quad \forall u \in V$$

The Optimal Power Flow (OPF) Problem

[Zimmerman et al.,
2011] $\in \mathbb{R}$

- Network $\mathcal{N}_{\text{bounded}} = (G, V_G, V_D, \text{cap}, \underline{b}, \underline{x}, \bar{x}, \underline{d}, \bar{d})$ is called bounded
- Generation cost function $\gamma_u: \mathbb{R} \rightarrow \mathbb{R}_{\geq 0}$ for all $u \in V_G$
- The OPTIMAL POWER FLOW (OPF) is denoted by $\text{OPF}(\mathcal{N})$ with value

$$\text{OPT}_{\text{OPF}}(\mathcal{N}) := \min \sum_{u \in V_G} \gamma_u(f_{\text{net}}(u))$$

with f being a **physical feasible** flow meaning

$$f_{\text{net}}(u) = 0 \quad \forall u \in V \setminus (V_G \cup V_D)$$

$$f_{\text{net}}(u) = -d_u \quad \forall u \in V_D$$

$$\underline{x} \leq f_{\text{net}}(u) \leq \bar{x} \quad \forall u \in V_G$$

$$|f(u, v)| \leq \text{cap}(u, v) \quad \forall (u, v) \in E$$

$$b(u, v) \cdot (\theta(v) - \theta(u)) = f(u, v) \quad \forall (u, v) \in E$$

$$\theta_{\min}(u) \leq \theta(u) \leq \theta_{\max}(u) \quad \forall u \in V$$

The Optimal Transmission Switching (OTS) Problem

[Fisher et al., 2008]

- The OPTIMAL TRANSMISSION SWITCHING (OTS) is denoted by

$$\text{OTS}(\mathcal{N}) := \min_{\mathcal{S} \subseteq \overleftrightarrow{E}} \text{OPF}(\mathcal{N} - \mathcal{S})$$

with value $\text{OPT}_{\text{OTS}}(\mathcal{N})$ with f being a **physical feasible** flow meaning

$$f_{\text{net}}(u) = 0 \quad \forall u \in V \setminus (V_G \cup V_D)$$

$$f_{\text{net}}(u) = -d_u \quad \forall u \in V_D$$

$$\underline{x} \leq f_{\text{net}}(u) \leq \bar{x} \quad \forall u \in V_G$$

$$|f(u, v)| \leq z(u, v) \cdot \text{cap}(u, v) \quad \forall (u, v) \in E$$

$$b(u, v) \cdot (\theta(v) - \theta(u)) + (1 - z(u, v))M \geq f(u, v) \quad \forall (u, v) \in E$$

$$b(u, v) \cdot (\theta(v) - \theta(u)) - (1 - z(u, v))M \leq f(u, v) \quad \forall (u, v) \in E$$

$$\theta_{\min}(u) \leq \theta(u) \leq \theta_{\max}(u) \quad \forall u \in V$$

The Optimal Transmission Switching (OTS) Problem

[Fisher et al., 2008]

Optimization Problem OTS

Instance: A power grid $\mathcal{N}_{\text{bounded}}$.

Objective: Find a set $\mathbf{S} \subseteq \underline{E}$ of switched edges such that $\text{OPT}_{\text{OPF}}(\mathcal{N} - \mathbf{S})$ is minimum among all choices of switched edges \mathbf{S} .

Decision Problem kOTS

Instance: A power grid $\mathcal{N}_{\text{bounded}}$ and $k \in \mathbb{Q}_{\geq 0}$.

Objective: Is it possible to remove a set of edges $\mathbf{S} \subseteq \underline{E}$ such that there is an physical feasible flow f in $\mathcal{N} - \mathbf{S}$ with cost $\sum_{u \in V_G} \gamma_u(f_{\text{net}}(u)) \leq k$?

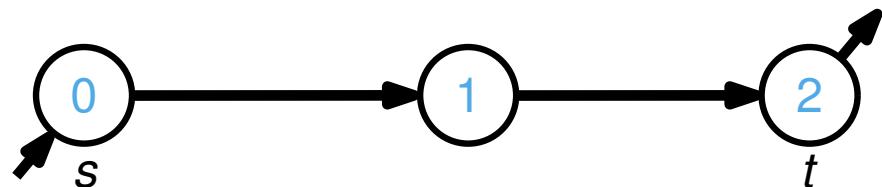
Connection to the DC-Model

- The power flow behaves like the current in the DC networks
- The total susceptance $B_{\text{tot}} = -\frac{1}{X}$ ($-\frac{1}{X}$, since $G = R = 0$ in DC-Approximation, otherwise $-\frac{X}{R^2+X^2}$) behaves like the electrical conductance $G = \frac{1}{R}$
- The voltage angle differences $\Delta\theta^\nu$ behave like the voltages U

Assumption: $b_k := 1, \text{cap}_k := 1 \quad \forall k \in \{1, \dots, m\}$

$$I_{\text{tot}} = \frac{U_{\text{tot}}}{R_{\text{tot}}} = U_{\text{tot}} \cdot G_{\text{tot}} \\ = \Delta\theta_{\text{tot}}^\nu \cdot B_{\text{tot}} = P_{\text{tot}}$$

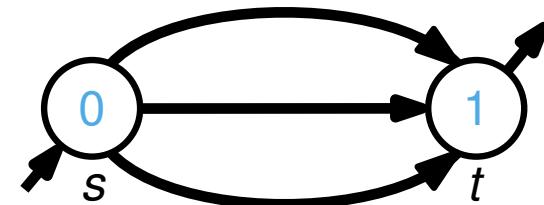
Series Circuit



$$B_{\text{tot}} = b(s, t) = \frac{1}{\sum_{k=1}^n \frac{1}{b_k}} = \frac{1}{2} \quad f(s, t) = 1$$

$$\Delta\theta_{\text{tot}}^\nu = \Delta\theta^\nu(s, t) = \frac{\text{cap}(s, t)}{b(s, t)} = 2$$

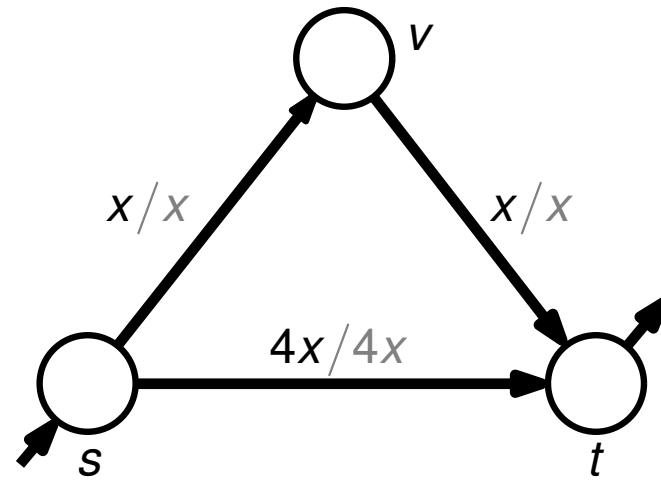
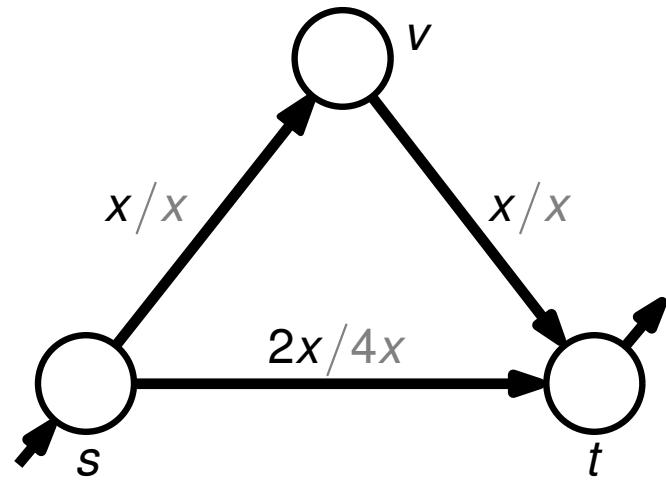
Parallel Circuit



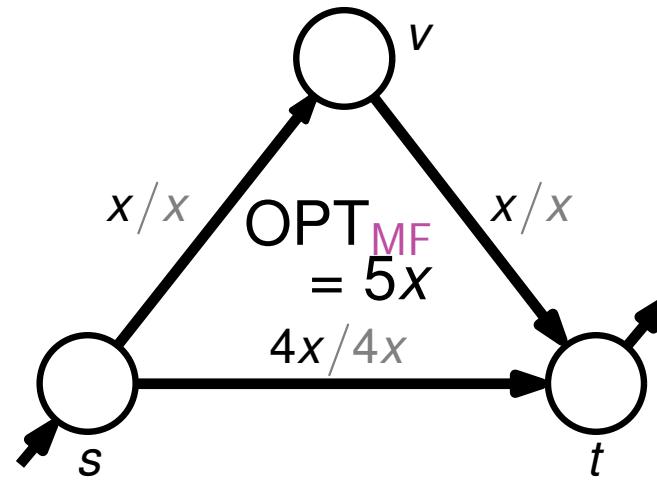
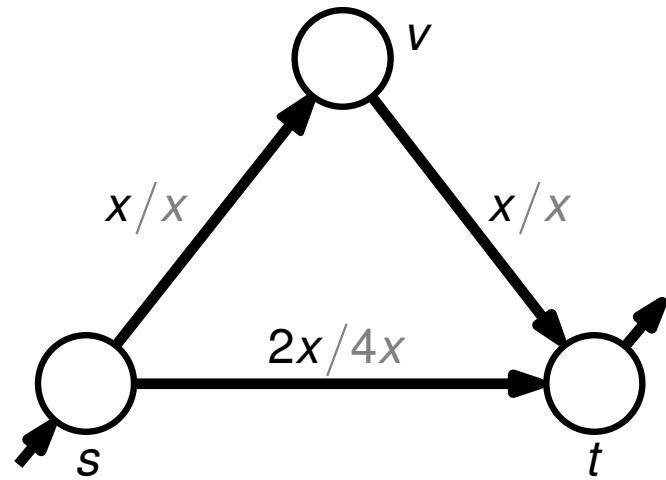
$$B_{\text{tot}} = b(s, t) = \sum_{k=1}^n b_k = 3 \quad f(s, t) = 3$$

$$\Delta\theta_{\text{tot}}^\nu = \Delta\theta^\nu(s, t) = \min_{(s,t)^i} (\Delta\theta_i^\nu(s, t)) = 1$$

The MTSF Problem



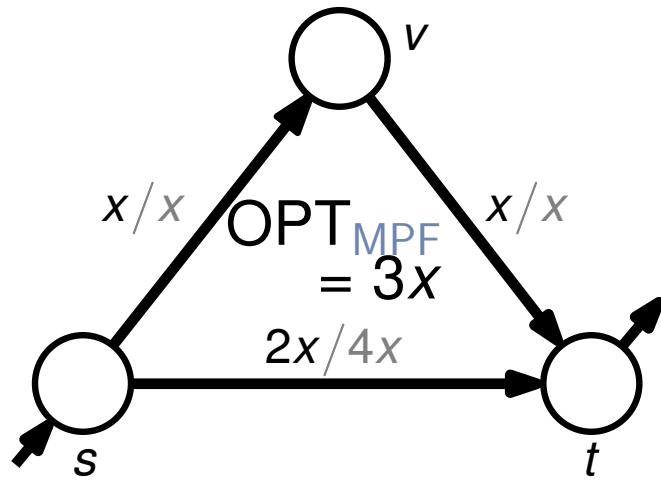
The MTSF Problem



flow model

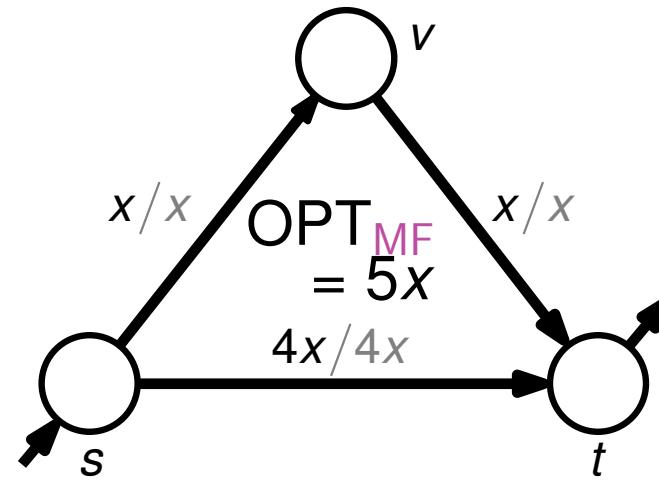
upper bound

The MTSF Problem



physical model
(AC linearization)

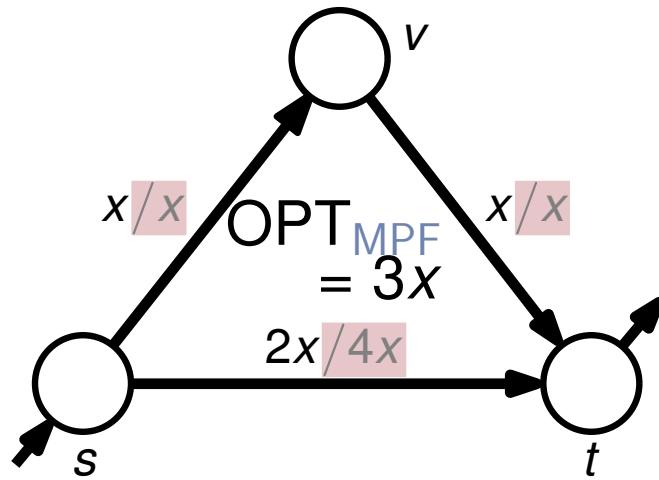
lower bound



flow model

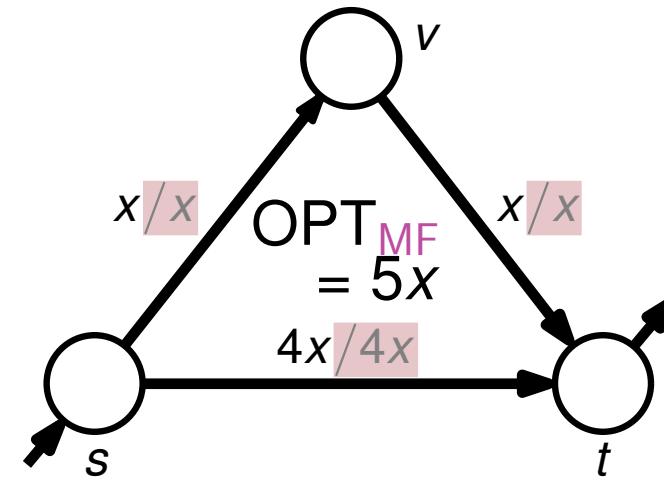
upper bound

The MTSF Problem



physical model
(AC linearization)

lower bound

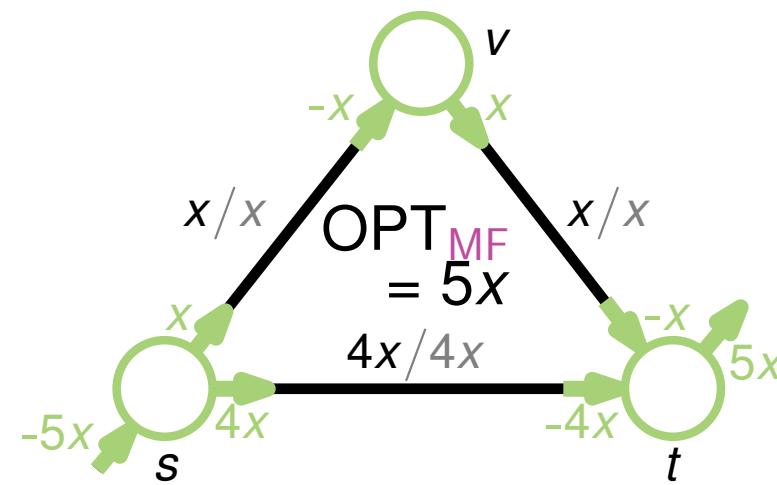
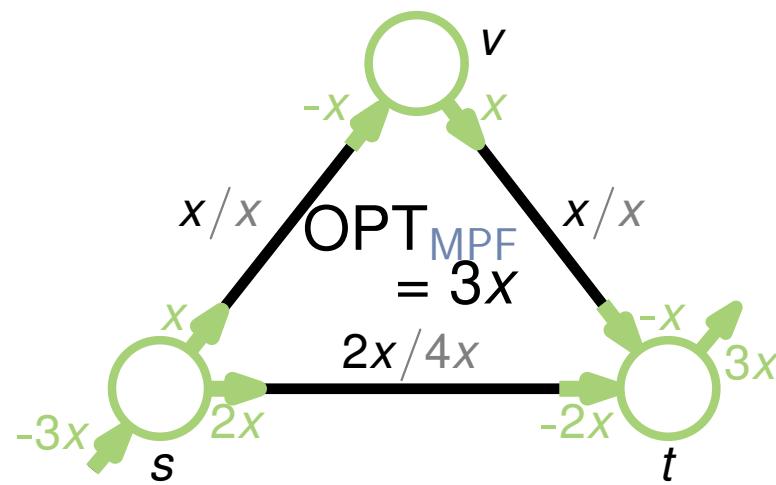


flow model

upper bound

capacity constraints

The MTSF Problem



physical model
(AC linearization)

lower bound

capacity constraints

Kirchhoff's Current Law (KCL)

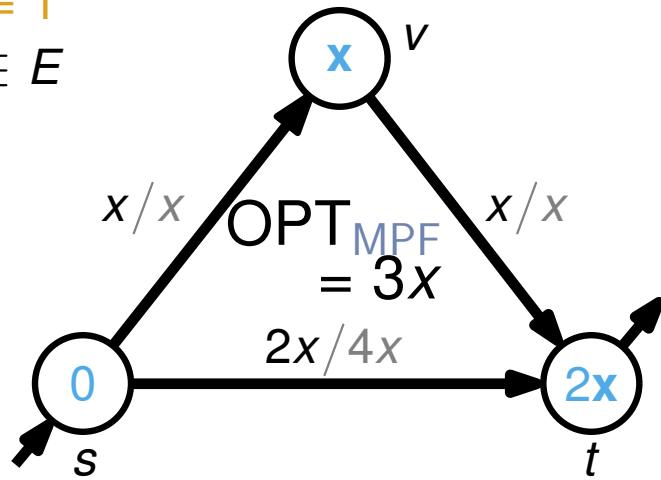
flow model

upper bound

The MTSF Problem

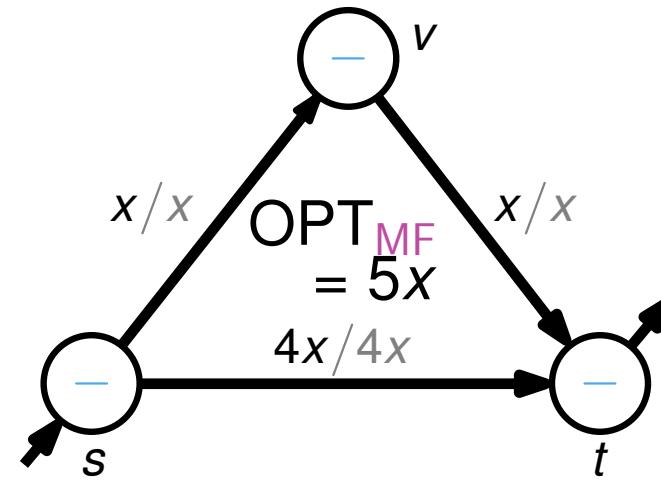
$$b(i, j) := 1$$

$$\forall (i, j) \in E$$



physical model
(AC linearization)

lower bound



flow model

upper bound

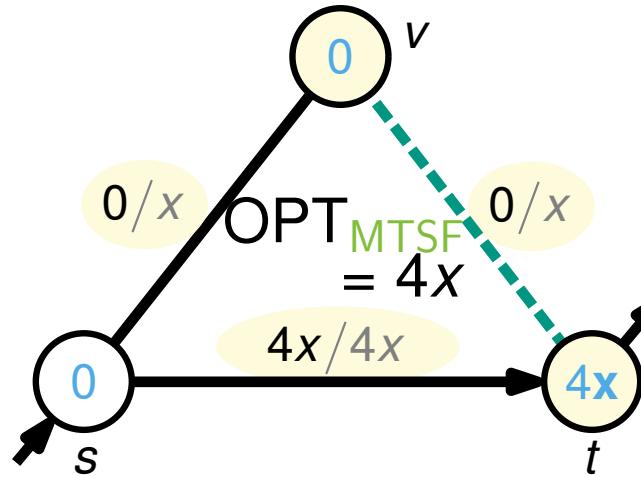
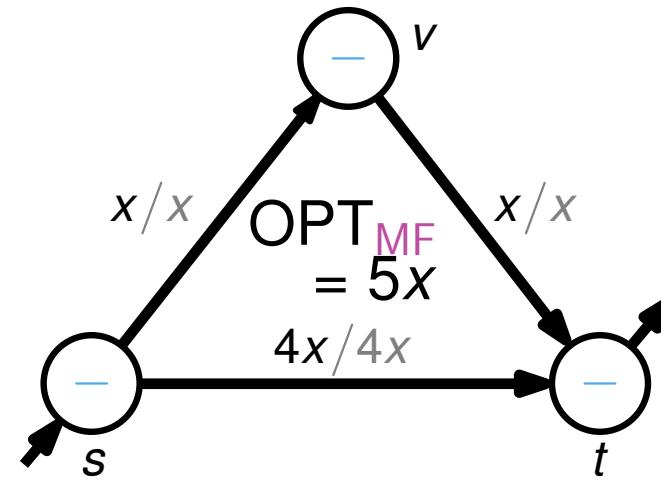
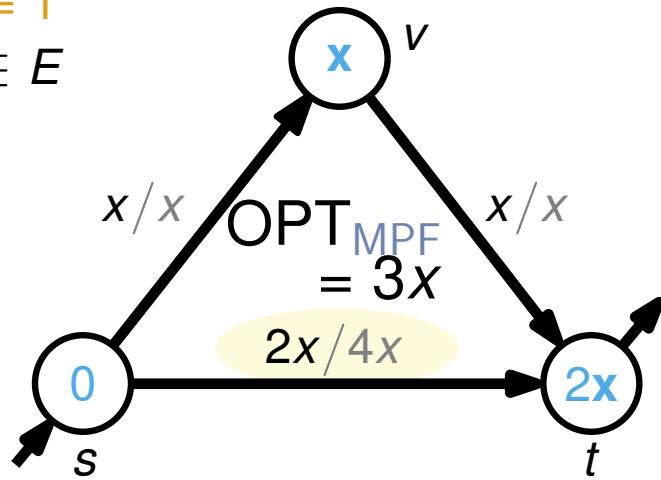
DC power flow constraints

$$\forall (u, v) \in E: f(u, v) = b(u, v)(\theta(v) - \theta(u))$$

The MTSF Problem

$$b(i, j) := 1$$

$$\forall (i, j) \in E$$

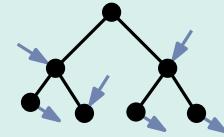
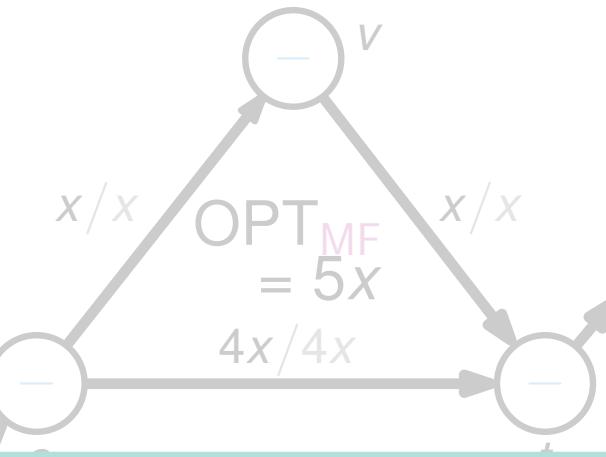
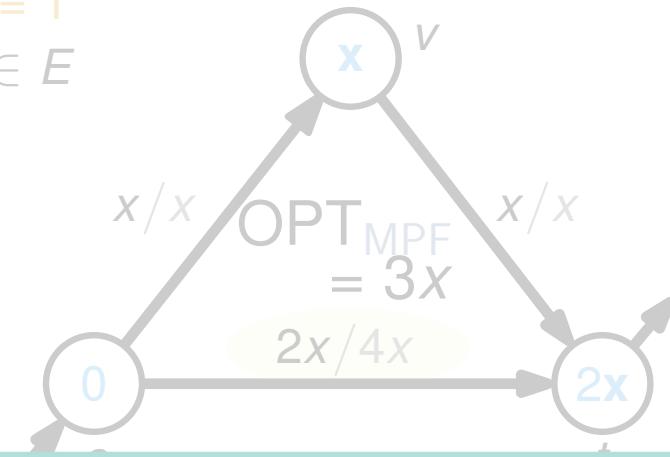


$$\forall (u, v) \in E: f(u, v) = b(u, v)(\theta(v) - \theta(u))$$

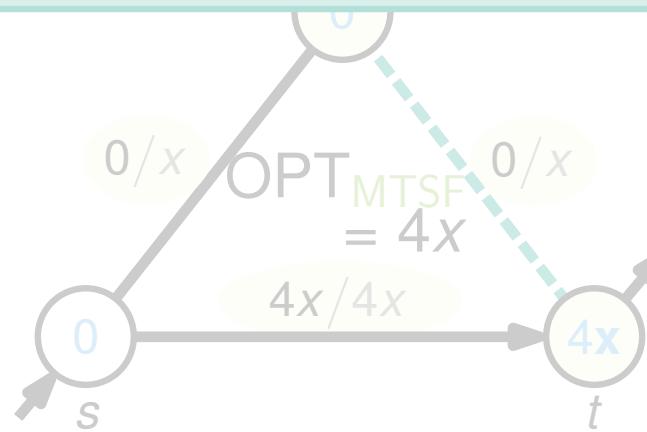
The MTSF Problem

$$b(i,j) := 1$$

$$\forall (i,j) \in E$$



Physical Model = Maximum Switching Flow = Flow Model (MPF)

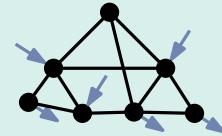
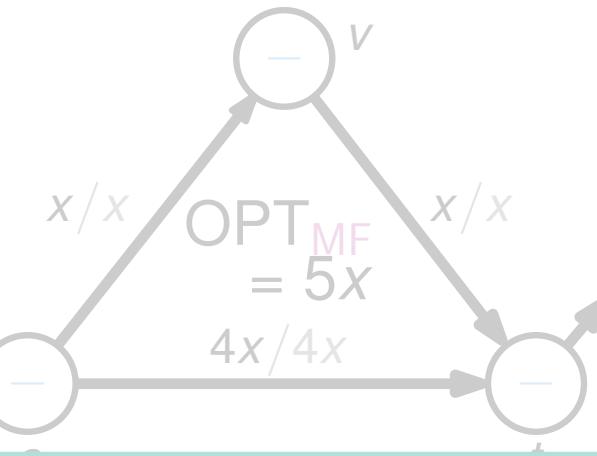
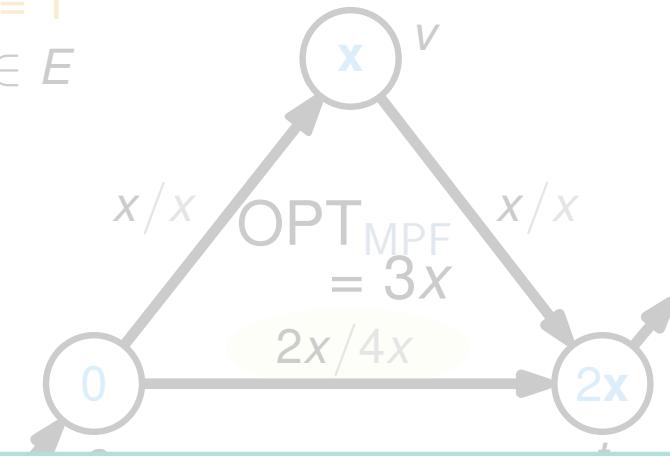


$$\forall (u, v) \in E: f(u, v) = b(u, v)(\theta(v) - \theta(u))$$

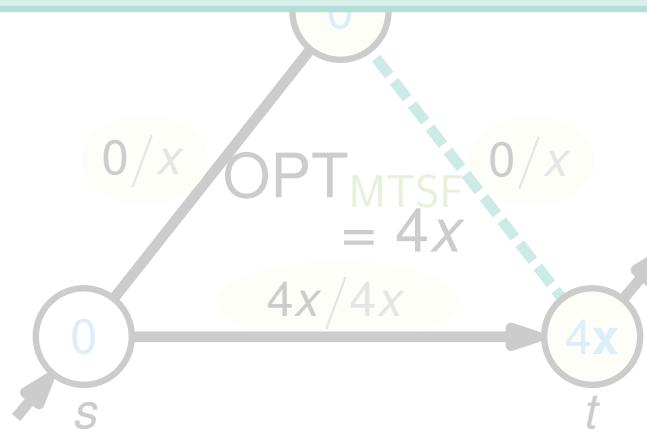
The MTSF Problem

$$b(i,j) := 1$$

$$\forall (i,j) \in E$$



Physical Model \leq Maximum Switching Flow \leq Flow Model
(MPF) \quad (MTSF) \quad (MF)



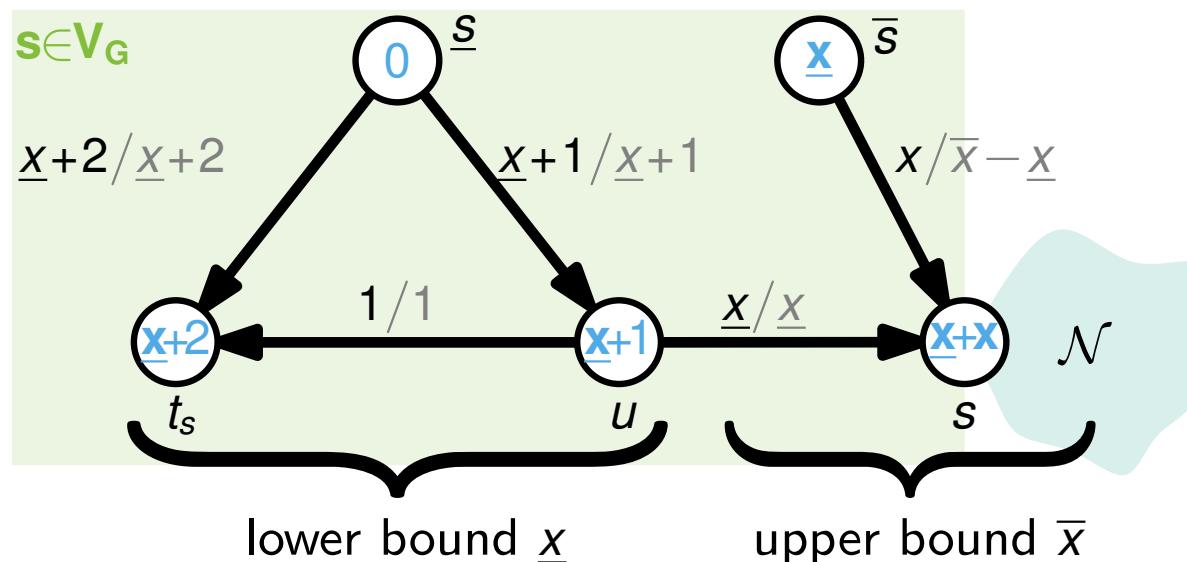
$$\forall (u, v) \in E: f(u, v) = b(u, v)(\theta(v) - \theta(u))$$

Network Modeling – Bounded to Unbounded

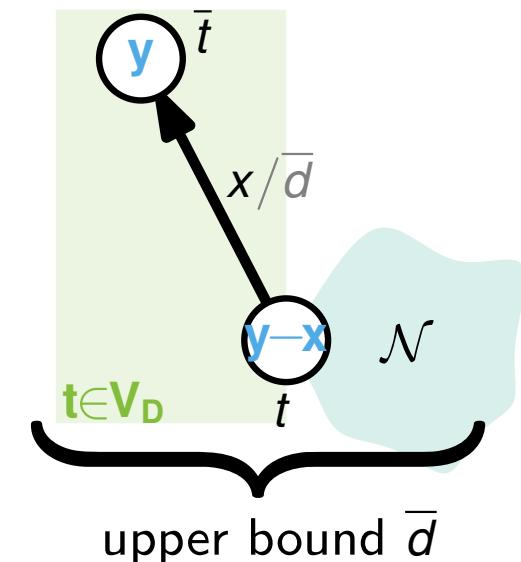
- Transformation from a bounded network

$\mathcal{N}_{\text{bounded}} = (G, V_G, V_D, \text{cap}, b, \underline{x}, \bar{x}, \underline{d}, \bar{d})$ to an unbounded network
 $\mathcal{N} = (G, V_G, V_D, \text{cap}, b, \underline{d})$

Model Generator Bounds in \mathcal{N}



Model Demand Bounds in \mathcal{N}



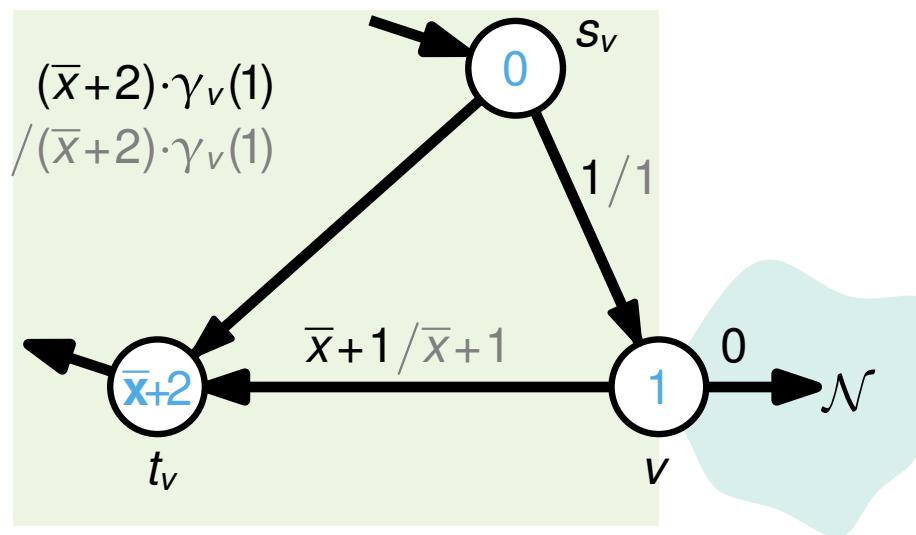
Lemma 1 [page 343; Grastien et al., 2018]

Every bounded MTSF can be transformed into an unbounded MTSF on a network with size linear in $|V|$ and $|E|$.

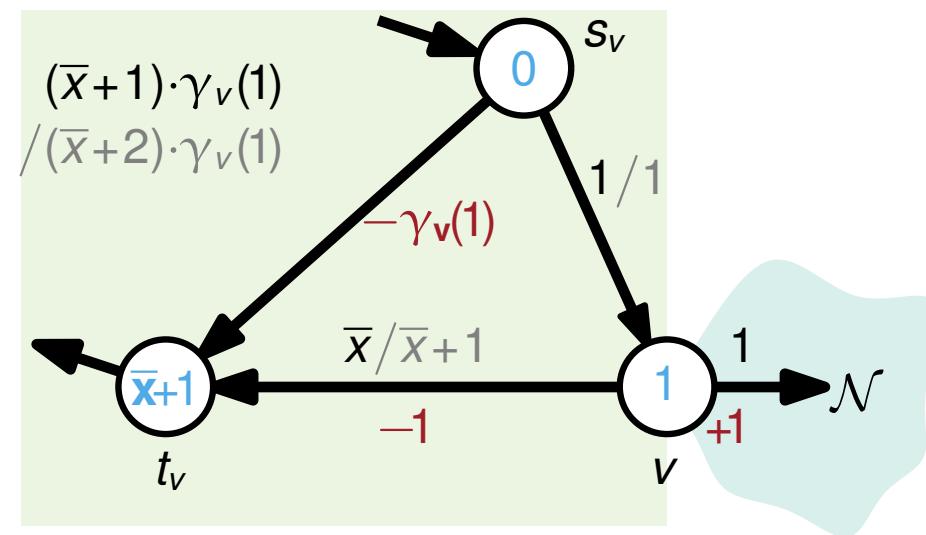
Network Modeling – OTS to MTSF

- OTS-instance $\mathcal{N} = (G, V_G, V_D, \text{cap}, \underline{b}, \underline{x}, \bar{x}, d)$
- $\gamma_v(1)$ cost per generated unit of power
- $b(s_v, t_v) := \gamma_v(1), b(s_v, v) := b(v, t_v) := 1$

Vertex v injects no power into network \mathcal{N}



Vertex v injects power into \mathcal{N}



- Feasible flow in \mathcal{N} with cost $k \rightarrow$ feasible flow in \mathcal{N}' with flow value $M - k$
- $M = \sum_{v \in V_G} ((\bar{x} + 2) \cdot \gamma_v(1) + \bar{x}_v + 1)$

Network Modeling – OTS to MTSF

- OTS-instance $\mathcal{N} = (G, V_G, V_D, \text{cap}, \underline{b}, \underline{x}, \bar{x}, d)$
- $\gamma_v(1)$ cost per generated unit of power
- $b(s_v, t_v) := \gamma_v(1)$, $b(s_v, v) := b(v, t_v) := 1$

Vertex v injects no power into network \mathcal{N}

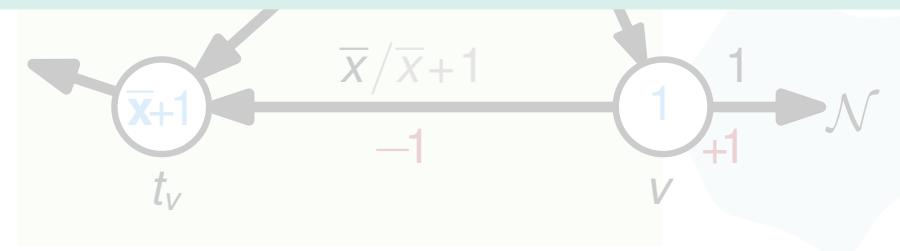


Vertex v injects power into \mathcal{N}



Lemma 2 [page 344; Grastien et al., 2018]

For every OTS-instance there is an MTSF-instance that is equivalent.



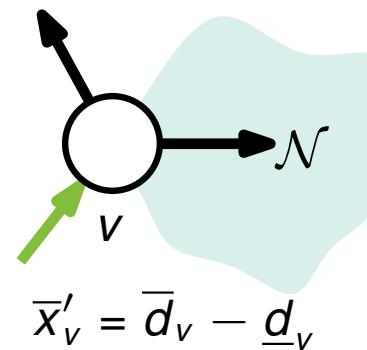
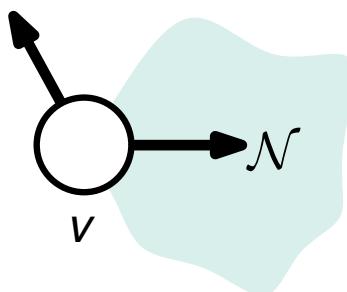
- Feasible flow in \mathcal{N} with cost $k \rightarrow$ feasible flow in \mathcal{N}' with flow value $M - k$
- $M = \sum_{v \in V_G} ((\bar{x} + 2) \cdot \gamma_v(1) + \bar{x}_v + 1)$

Network Modeling – MTSF to OTS

[Lehmann et al.,
2014]

- MTSF-instance $\mathcal{N} = (G, V_G, V_D, \text{cap}, \underline{b}, \bar{x}, \underline{d}, \bar{d})$ with $V_G \cap V_D = \emptyset$
- OTS-instance $\mathcal{N}' = (G, V'_G, V_D, \text{cap}, \underline{b}, \bar{x}', d' = \underline{d}' = \bar{d}, \gamma)$
- The new generator set $V'_G = V_G \cup \{v \in V_D | \bar{d}_v > 0\}$
- $\gamma_v(1) = 1$ cost per generated unit of power

$$\sum_{v \in V} \bar{d}_v - \text{MTSF}(\mathcal{N}) = \text{OTS}(\mathcal{N}')$$



$v \in (V_D \cap V'_G)$

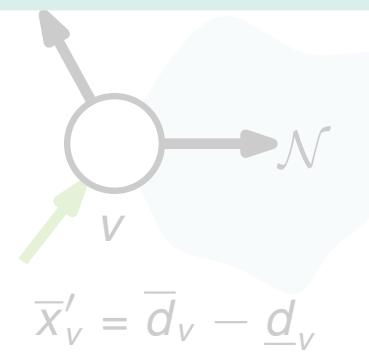
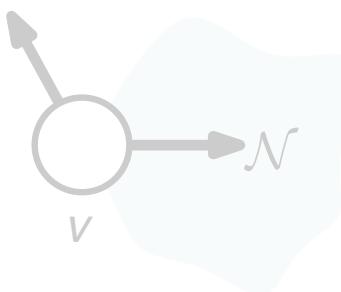
Network Modeling – MTSF to OTS

[Lehmann et al.,
2014]

- MTSF-instance $\mathcal{N} = (G, V_G, V_D, \text{cap}, b, \bar{x}, \underline{d}, \bar{d})$ with $V_G \cap V_D = \emptyset$
- OTS-instance $\mathcal{N}' = (G, V'_G, V_D, \text{cap}, b, \bar{x}', d' = \underline{d}' = \bar{d}, \gamma)$
- The new generator set $V'_G = V_G \cup \{v \in V_D | \bar{d}_v > 0\}$
- $\gamma_v(1) = 1$ cost per generated unit of power

Lemma 3 [page 6; Lehmann et al., 2014]

For every MTSF-instance there is an OTS-instance that is equivalent.



$v \in (V_D \cap V'_G)$

$$\bar{x}'_v = \bar{d}_v - \underline{d}_v$$

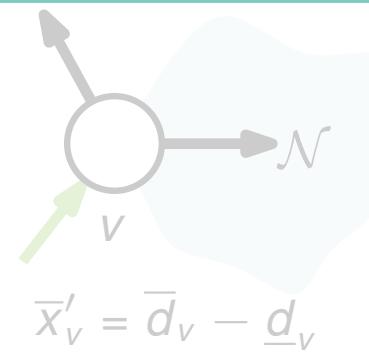
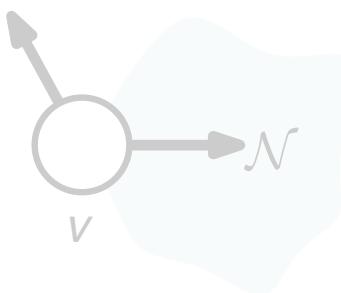
Network Modeling – MTSF to OTS

[Lehmann et al.,
2014]

- MTSF-instance $\mathcal{N} = (G, V_G, V_D, \text{cap}, \underline{b}, \bar{x}, \underline{d}, \bar{d})$ with $V_G \cap V_D = \emptyset$
- OTS-instance $\mathcal{N}' = (G, V'_G, V_D, \text{cap}, \underline{b}, \bar{x}', d' = \underline{d}' = \bar{d}, \gamma)$
- The new generator set $V'_G = V_G \cup \{v \in V_D | \bar{d}_v > 0\}$
- $\gamma_v(1) = 1$ cost per generated unit of power

$$\sum_{v \in V_D} \bar{d}_v = \text{MTSF}(\mathcal{N}) = \text{OTS}(\mathcal{N}')$$

Any hardness result for MTSF holds also for OTS. This does not hold for approximation results.

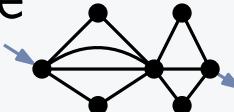


$v \in (V_D \cap V'_G)$

Overview of the MTSF Results

Graph Structure	Complexity	Algorithm

Overview of the MTSF Results

Graph Structure	Complexity	Algorithm
<p>penrose-minor-free graphs</p> <p>easy One generator, one load</p> 	polynomial-time solvable	DTP

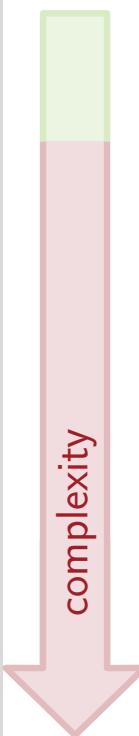
Overview of the MTSF Results

Graph Structure	Complexity	Algorithm
penrose-minor-free graphs series-parallel graphs	polynomial-time solvable NP-hard	DTP ✓ X

Overview of the MTSF Results

Graph Structure	Complexity	Algorithm
penrose-minor-free graphs series-parallel graphs cacti with max degree of 3	polynomial-time solvable NP-hard NP-hard [Lehmann et al. 2014]	DTP X 2-approx.
one generator, one load		
arbitrary generators, arbitrary loads		

Overview of the MTSF Results

Graph Structure	Complexity	Algorithm
 <div style="background-color: #e0e0ff; padding: 5px; display: inline-block;"> one generator, one load </div> penrose-minor-free graphs series-parallel graphs cacti with max degree of 3 2-level trees	polynomial-time solvable	 DTP
	NP-hard	
	NP-hard <small>[Lehmann et al. 2014]</small>	 2-approx.
	NP-hard <small>[Lehmann et al. 2014]</small>	

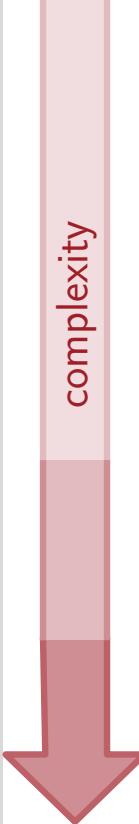
Overview of the MTSF Results

Graph Structure	Complexity	Algorithm
penrose-minor-free graphs series-parallel graphs cacti with max degree of 3 2-level trees planar graphs with max degree of 3	one generator, one load arbitrary generators, arbitrary loads	DTP
	polynomial-time solvable	
	NP-hard	
	NP-hard <small>[Lehmann et al. 2014]</small>	2-approx.
	NP-hard <small>[Lehmann et al. 2014]</small>	
	strongly NP-hard <small>[Lehmann et al. 2014]</small>	

Overview of the MTSF Results

	Graph Structure	Complexity	Algorithm
complexity ↓	penrose-minor-free graphs	polynomial-time solvable	DTP
	series-parallel graphs	NP-hard	
	cacti with max degree of 3	NP-hard <small>[Lehmann et al. 2014]</small>	2-approx.
	2-level trees	NP-hard <small>[Lehmann et al. 2014]</small>	
	planar graphs with max degree of 3	strongly NP-hard <small>[Lehmann et al. 2014]</small>	
	arbitrary graphs	non-APX <small>[Lehmann et al. 2014]</small>	

Overview of the MTSF Results

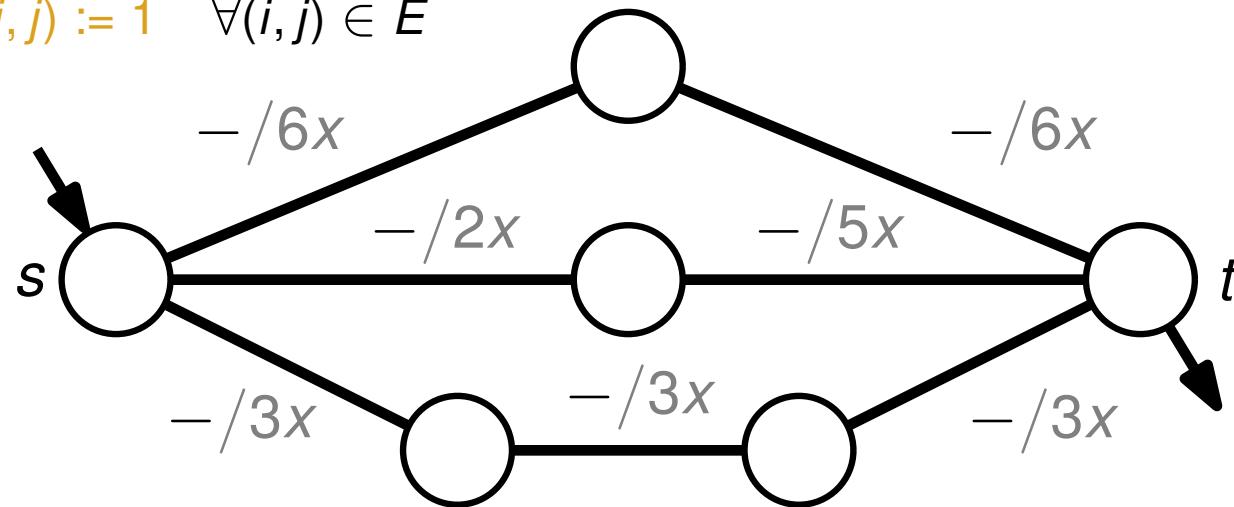
	Graph Structure	Complexity	Algorithm
complexity 	penrose-minor-free graphs	polynomial-time solvable	DTP 
	series-parallel graphs	NP-hard	
	cacti with max degree of 3	NP-hard [Lehmann et al., 2014]	2-approx. 
	2-level trees	NP-hard [Lehmann et al., 2014]	
	planar graphs with max degree of 3	strongly NP-hard [Lehmann et al., 2014]	
	arbitrary graphs	non-APX [Lehmann et al., 2014]	

Overview of the MTSF Results

	Graph Structure	Complexity	Algorithm
complexity ↓ $ V_G = 2, V_D = 2$	penrose-minor-free graphs	polynomial-time solvable	DTP
	series-parallel graphs	NP-hard	
	cacti with max degree of 3	NP-hard [Lehmann et al., 2014]	2-approx.
	2-level trees	NP-hard [Lehmann et al., 2014]	
	planar graphs with max degree of 3	strongly NP-hard [Lehmann et al., 2014]	
	arbitrary graphs	non-APX [Lehmann et al., 2014]	

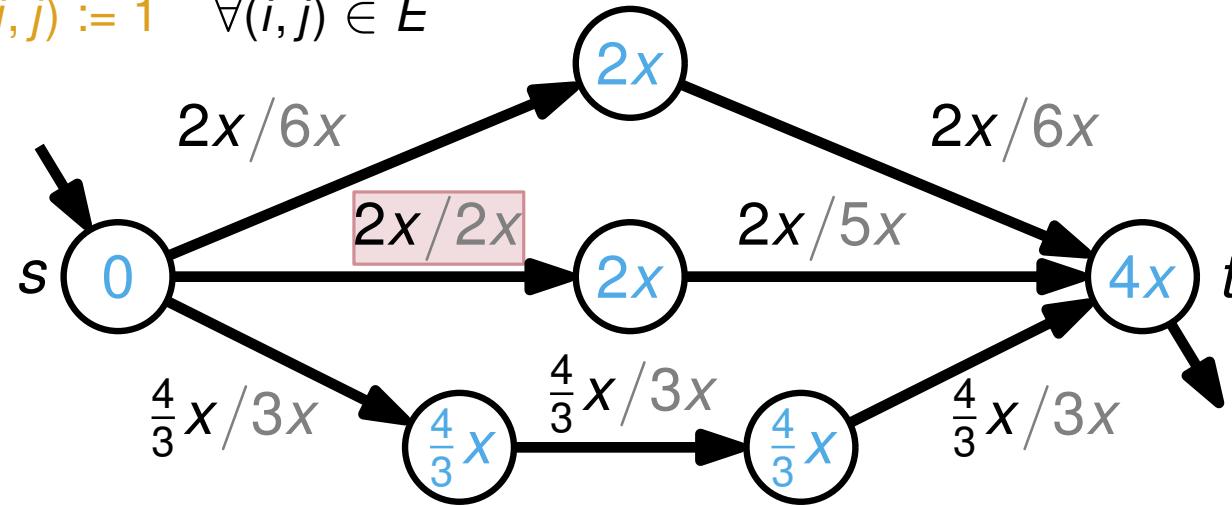
Switching on Parallel Paths

$$b(i, j) := 1 \quad \forall (i, j) \in E$$



Switching on Parallel Paths

$$b(i, j) := 1 \quad \forall (i, j) \in E$$



$$F = \frac{16}{3}x \approx 5.33x$$

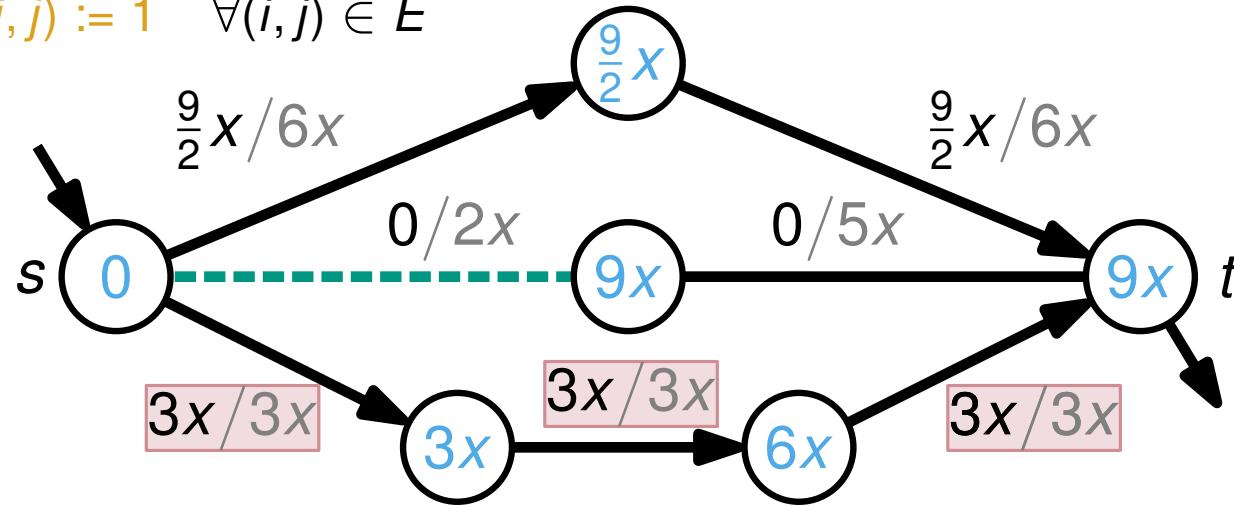
$$\Delta\theta^v(\pi) := \|\pi\|_b \cdot \min_{e \in \pi} \text{cap}(e)$$

electrical distance

$$\Delta\theta^v(\pi_2) = 2 \cdot 2x = 4x$$

Switching on Parallel Paths

$$b(i, j) := 1 \quad \forall (i, j) \in E$$



$$\Delta\theta^v(\pi) := \underbrace{\|\pi\|_b}_{\text{electrical distance}} \cdot \min_{e \in \pi} \text{cap}(e)$$

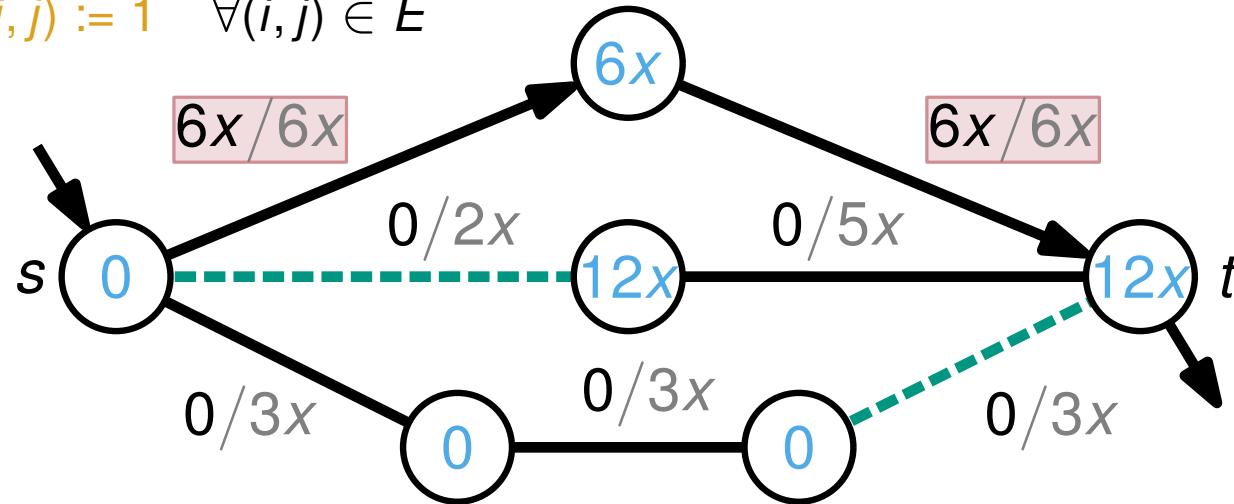
$$\Delta\theta^v(\pi_2) = 2 \cdot 2x = 4x$$

$$\Delta\theta^v(\pi_3) = 3 \cdot 3x = 9x$$

$$F = \frac{15}{2}x = 7.5x$$

Switching on Parallel Paths

$$b(i, j) := 1 \quad \forall (i, j) \in E$$



$$F = 6x$$

$$\Delta\theta^v(\pi) := \underbrace{\|\pi\|_b}_{\text{electrical distance}} \cdot \min_{e \in \pi} \text{cap}(e)$$

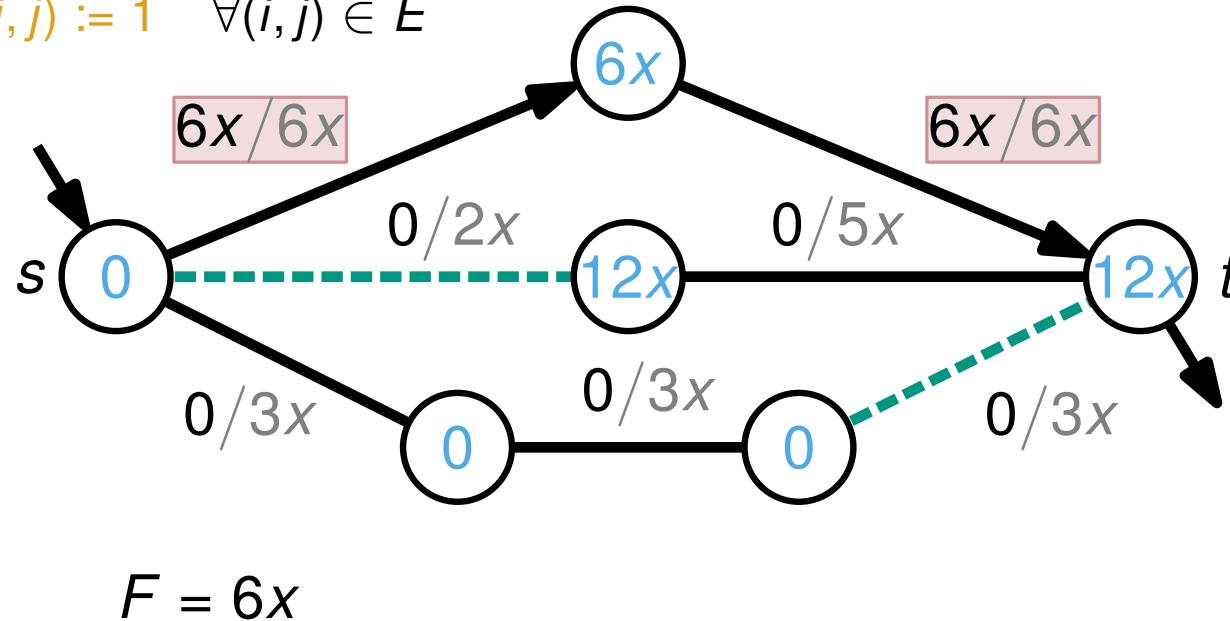
$$\Delta\theta^v(\pi_1) = 2 \cdot 6x = 12x$$

$$\Delta\theta^v(\pi_2) = 2 \cdot 2x = 4x$$

$$\Delta\theta^v(\pi_3) = 3 \cdot 3x = 9x$$

Switching on Parallel Paths

$$b(i, j) := 1 \quad \forall (i, j) \in E$$



$$\Delta\theta^v(\pi) := \|\pi\|_b \cdot \min_{e \in \pi} \text{cap}(e)$$

electrical distance

$$\Delta\theta^v(\pi_1) = 2 \cdot 6x = 12x$$

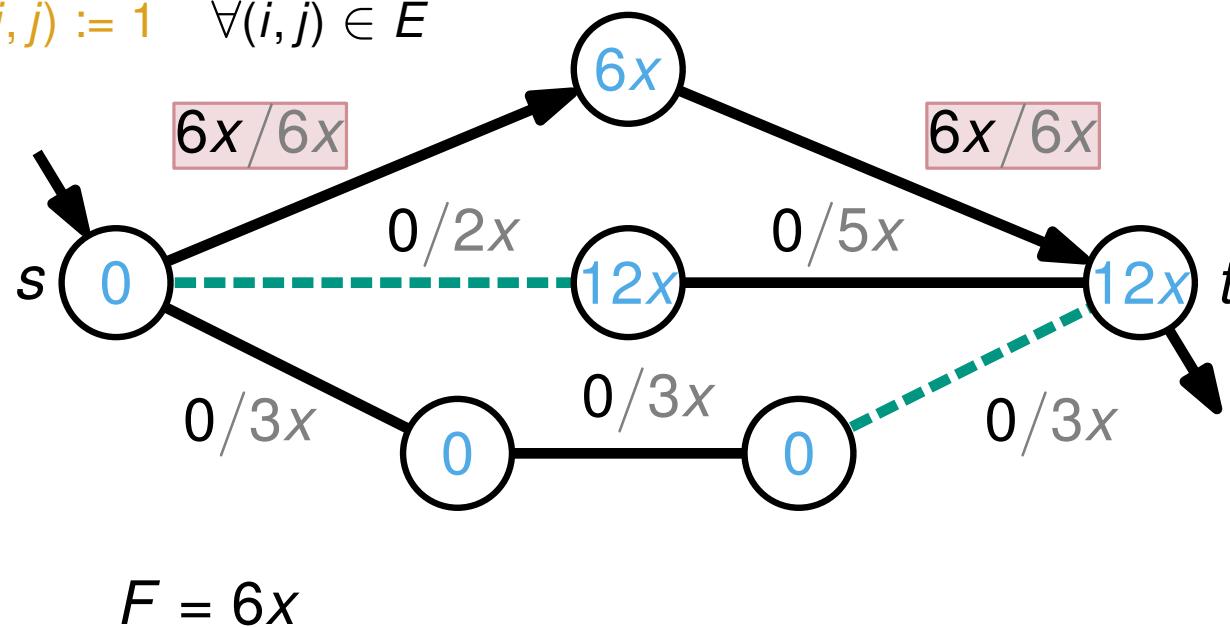
$$\Delta\theta^v(\pi_2) = 2 \cdot 2x = 4x$$

$$\Delta\theta^v(\pi_3) = 3 \cdot 3x = 9x$$

On parallel paths an optimal switching can be computed in polynomial time.

Switching on Parallel Paths

$$b(i, j) := 1 \quad \forall (i, j) \in E$$



$$\Delta\theta^v(\pi) := \underbrace{\|\pi\|_b}_{\text{electrical distance}} \cdot \min_{e \in \pi} \text{cap}(e)$$

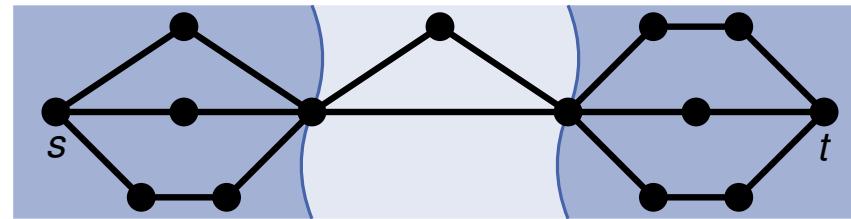
$$\Delta\theta^v(\pi_1) = 2 \cdot 6x = 12x$$

$$\Delta\theta^v(\pi_2) = 2 \cdot 2x = 4x$$

$$\Delta\theta^v(\pi_3) = 3 \cdot 3x = 9x$$

On parallel paths an optimal switching can be computed in polynomial time.

The same holds for series compositions of these graphs.



Dominating Theta Path (**DTP**)

[Section 5; Grastien et al., 2018]

Fix $u, v \in V$ and a u - v -path π .

Susceptance Norm:

$$\|\pi\|_b := \sum_{e \in E(\pi)} \frac{1}{b(e)}$$

Minimum Capacity:

$$\underline{\text{cap}}(\pi) := \min\{\text{cap}(e) \mid e \in \pi\}$$

Dominating Theta Path (DTP)

[Section 5; Grastien et al., 2018]

Fix $u, v \in V$ and a u - v -path π .

Susceptance Norm:

$$\|\pi\|_b := \sum_{e \in E(\pi)} \frac{1}{b(e)}$$

Minimum Capacity:

$$\underline{\text{cap}}(\pi) := \min\{\text{cap}(e) \mid e \in \pi\}$$

Angle Difference of π :

$$\Delta\theta^v(\pi) := \|\pi\|_b \cdot \underline{\text{cap}}(\pi)$$

Dominating Theta Path (DTP)

[Section 5; Grastien et al., 2018]

Fix $u, v \in V$ and a u - v -path π .

Susceptance Norm:

$$\|\pi\|_b := \sum_{e \in E(\pi)} \frac{1}{b(e)}$$

Minimum Capacity:

$$\underline{\text{cap}}(\pi) := \min\{\text{cap}(e) \mid e \in \pi\}$$

Angle Difference of π :

$$\Delta\theta^v(\pi) := \|\pi\|_b \cdot \underline{\text{cap}}(\pi)$$

Dominating Theta Path (DTP):

$$\Delta\theta_{\min}^v(u, v) := \min\{\Delta\theta^v(\pi) \mid \pi \text{ is a } u\text{-}v\text{-path}\}$$

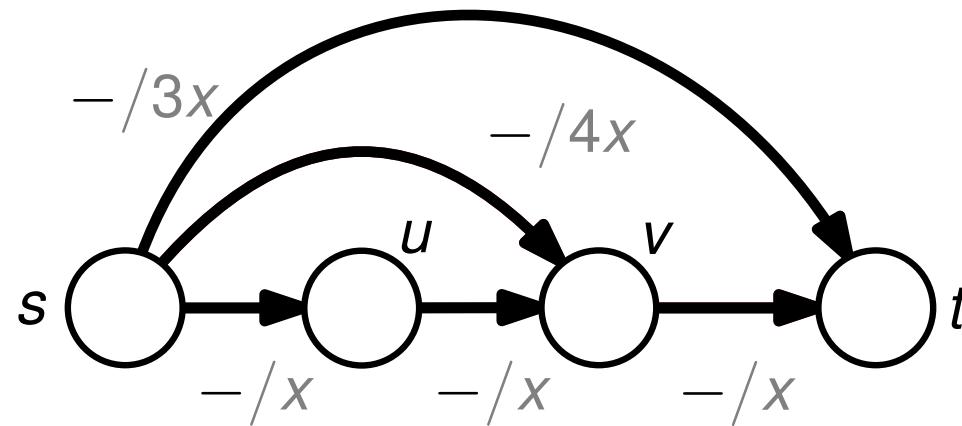
Computing DTP

[Section 5; Grastien et al., 2018]

Description:

- Bicriterial Dijkstra with labels $(\|\pi\|_b, \text{cap}(\pi), V_i)$
- at most $|E|$ labels per vertex

$$b(i, j) := 1 \quad \forall (i, j) \in E$$



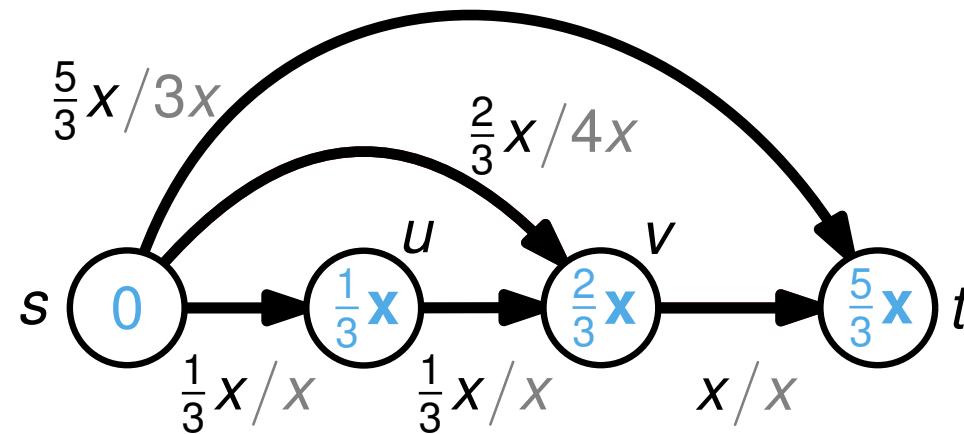
Computing DTP

[Section 5; Grastien et al., 2018]

Description:

- Bicriterial Dijkstra with labels $(\|\pi\|_b, \text{cap}(\pi), V_i)$
- at most $|E|$ labels per vertex

$$b(i, j) := 1 \quad \forall (i, j) \in E$$



$$\text{OPT}_{\text{MPF}} = \frac{8}{3}x$$

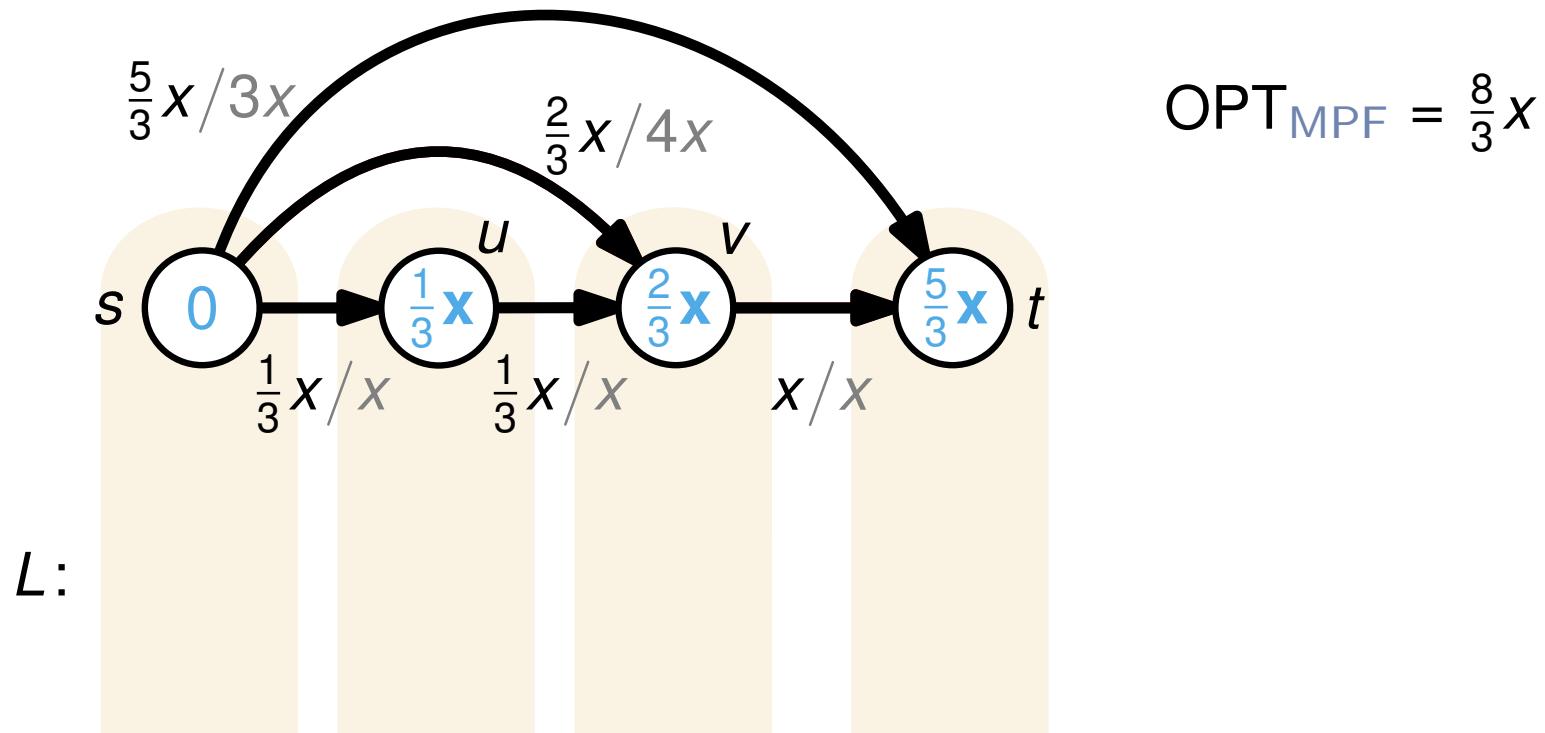
Computing DTP

[Section 5; Grastien et al., 2018]

Description:

- Bicriterial Dijkstra with labels $(\|\pi\|_b, \text{cap}(\pi), V_i)$
- at most $|E|$ labels per vertex

$$b(i, j) := 1 \quad \forall (i, j) \in E$$



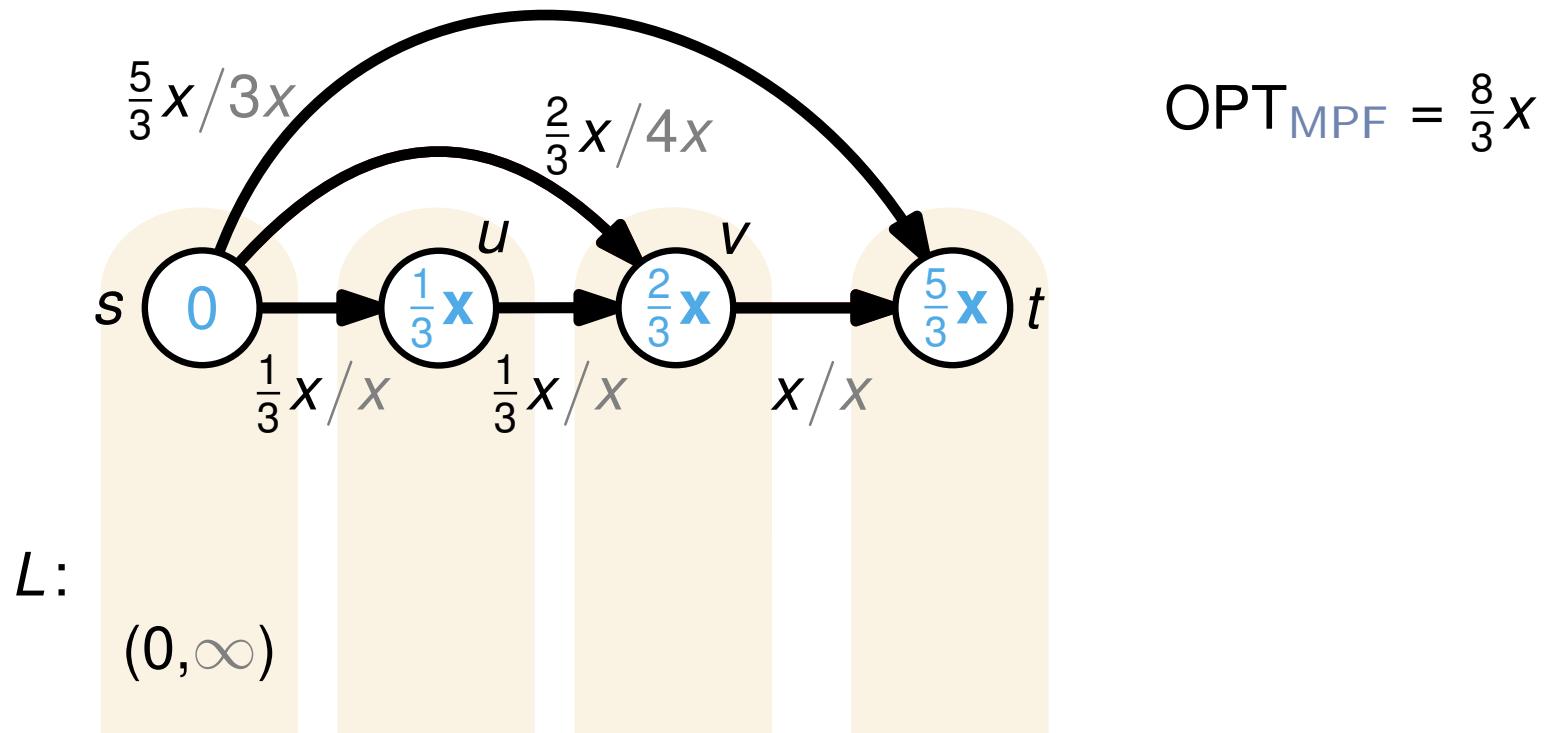
Computing DTP

[Section 5; Grastien et al., 2018]

Description:

- Bicriterial Dijkstra with labels $(\|\pi\|_b, \text{cap}(\pi), V_i)$
- at most $|E|$ labels per vertex

$$b(i, j) := 1 \quad \forall (i, j) \in E$$



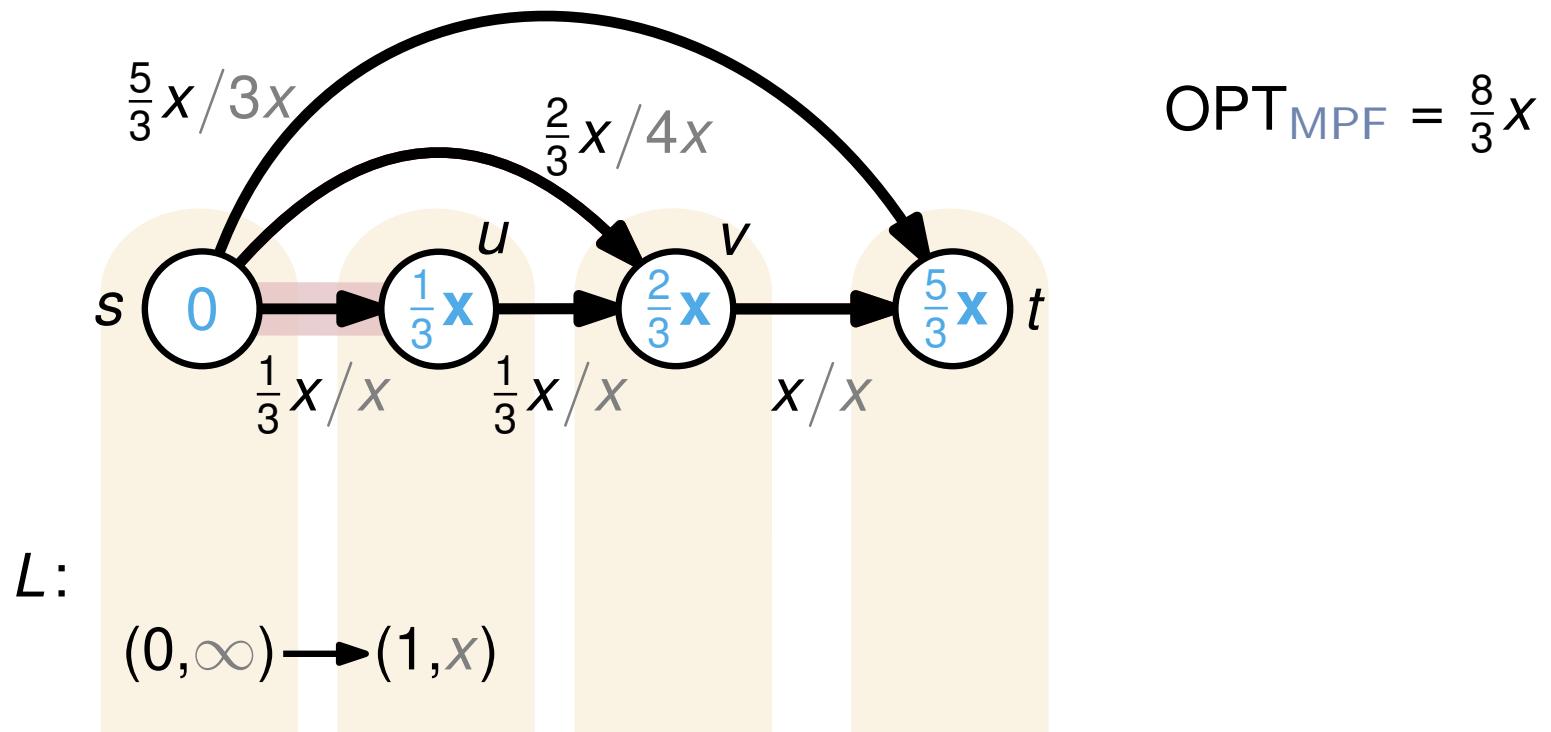
Computing DTP

[Section 5; Grastien et al., 2018]

Description:

- Bicriterial Dijkstra with labels $(\|\pi\|_b, \text{cap}(\pi), V_i)$
- at most $|E|$ labels per vertex

$$b(i, j) := 1 \quad \forall (i, j) \in E$$



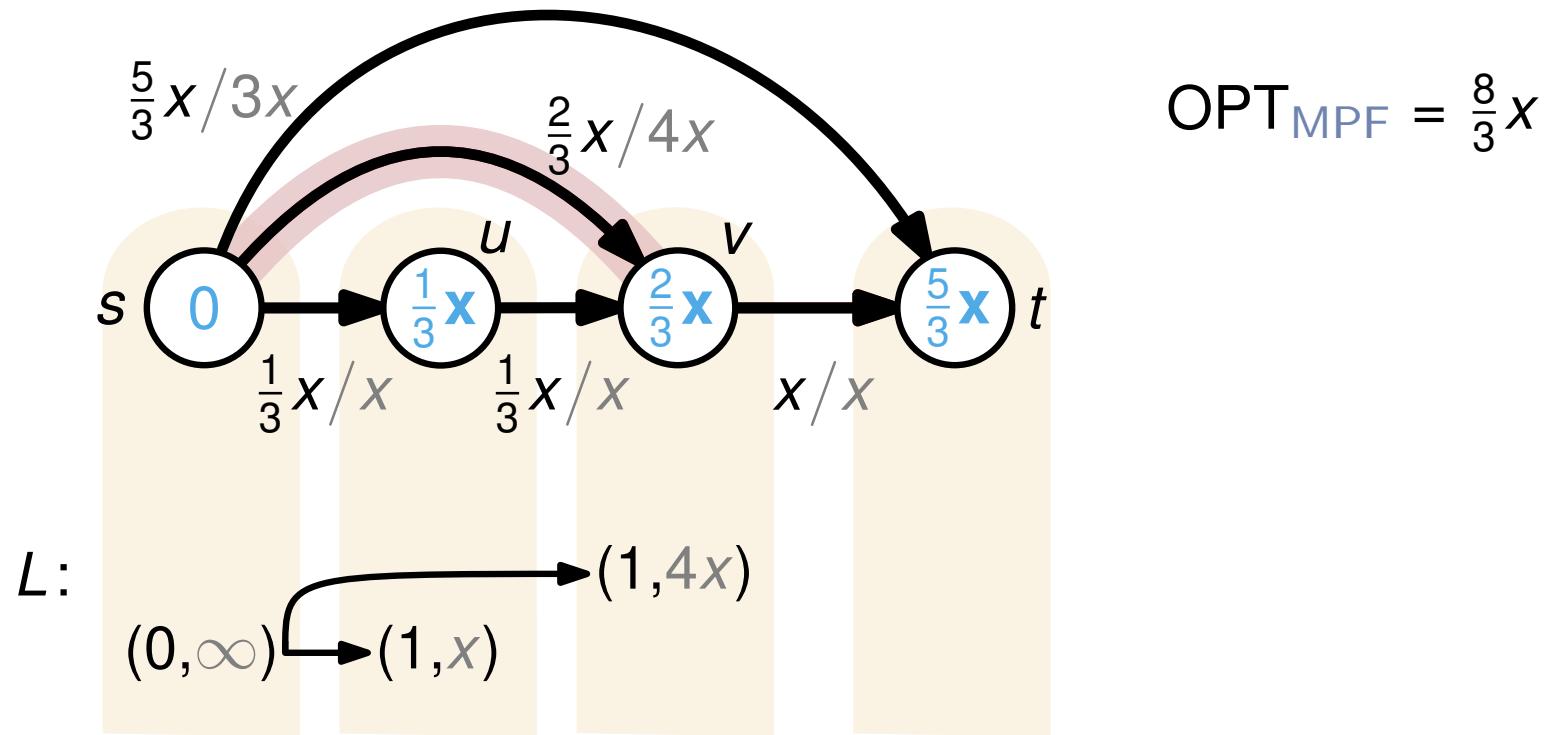
Computing DTP

[Section 5; Grastien et al., 2018]

Description:

- Bicriterial Dijkstra with labels $(\|\pi\|_b, \text{cap}(\pi), V_i)$
- at most $|E|$ labels per vertex

$$b(i, j) := 1 \quad \forall (i, j) \in E$$



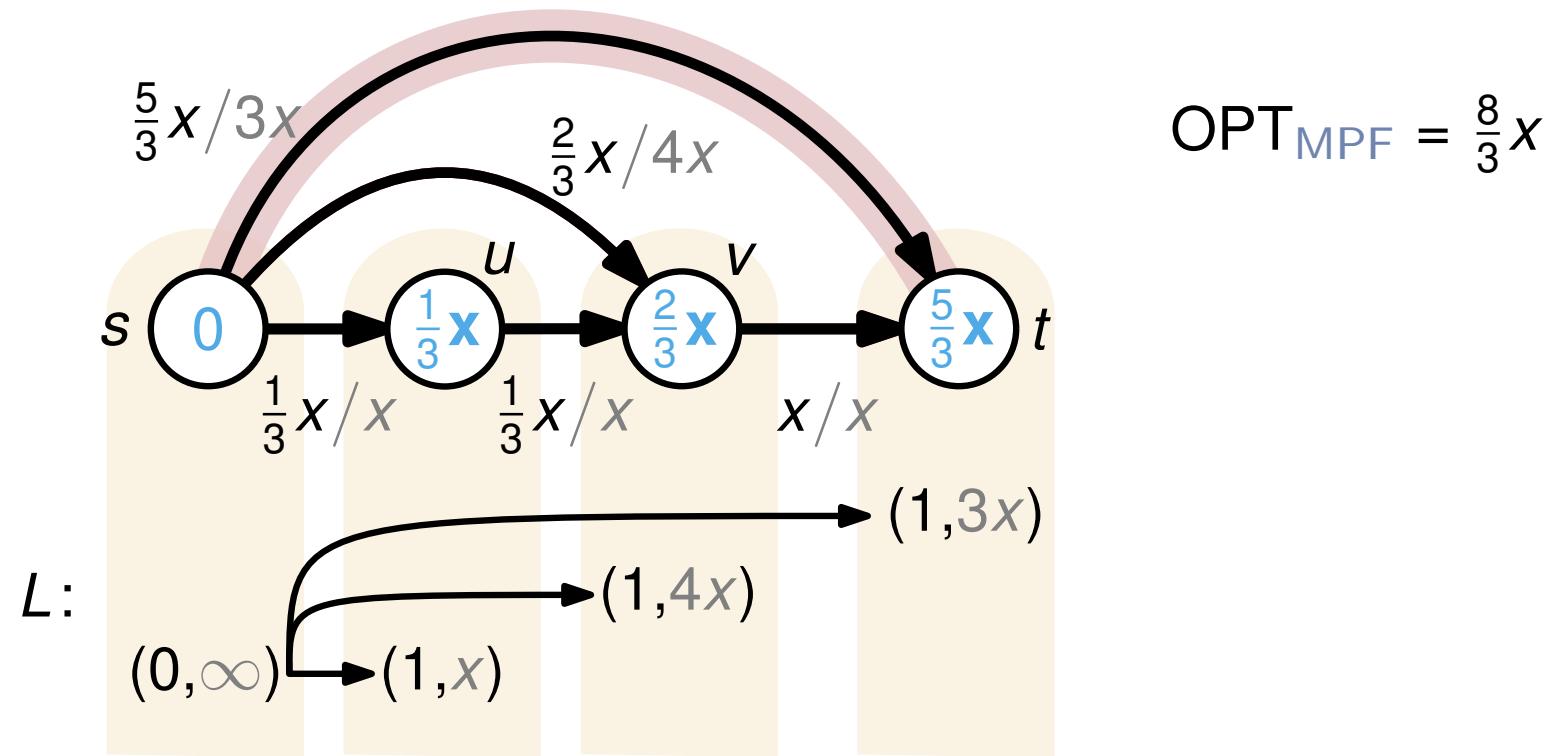
Computing DTP

[Section 5; Grastien et al., 2018]

Description:

- Bicriterial Dijkstra with labels $(\|\pi\|_b, \text{cap}(\pi), V_i)$
- at most $|E|$ labels per vertex

$$b(i, j) := 1 \quad \forall (i, j) \in E$$



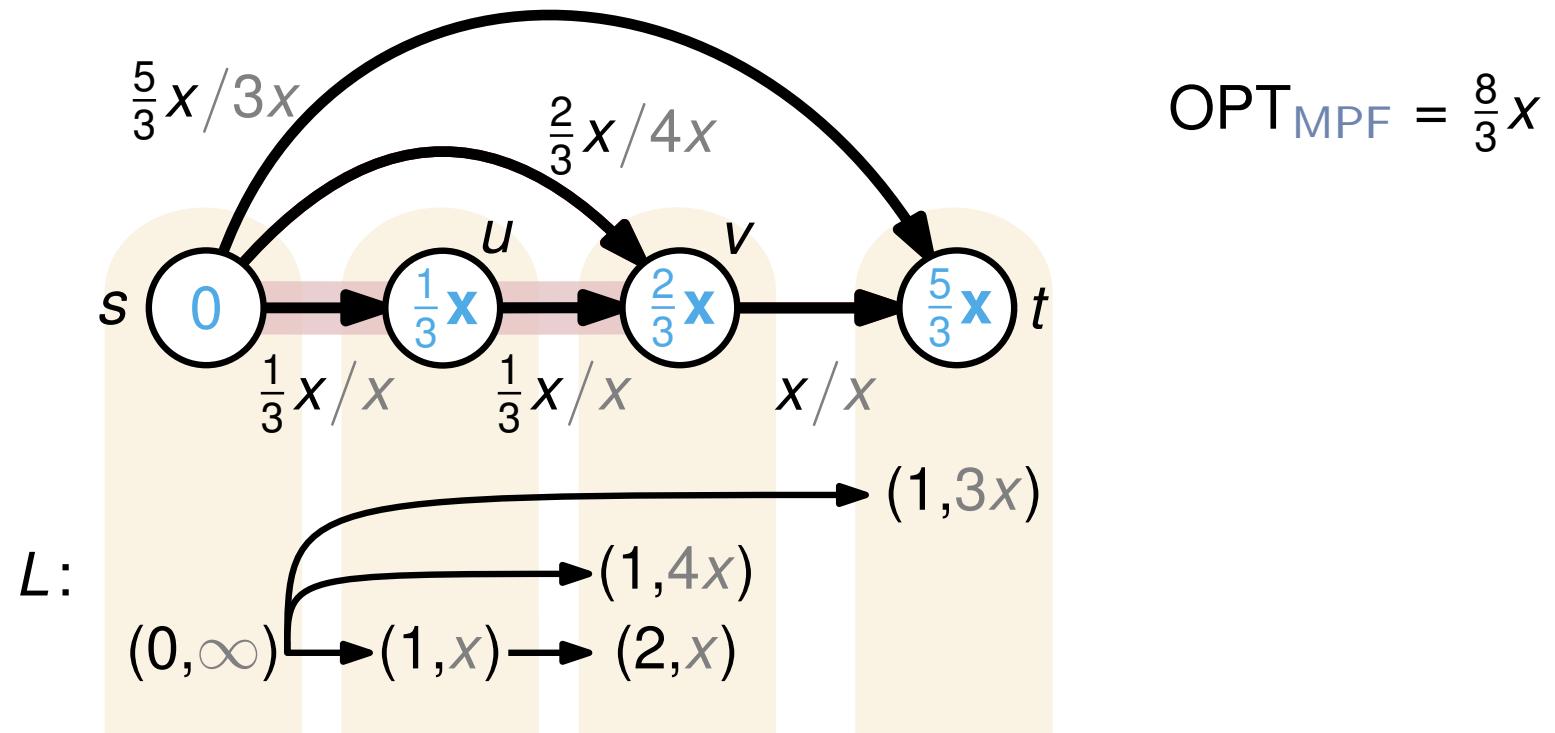
Computing DTP

[Section 5; Grastien et al., 2018]

Description:

- Bicriterial Dijkstra with labels $(\|\pi\|_b, \text{cap}(\pi), V_i)$
- at most $|E|$ labels per vertex

$$b(i, j) := 1 \quad \forall (i, j) \in E$$



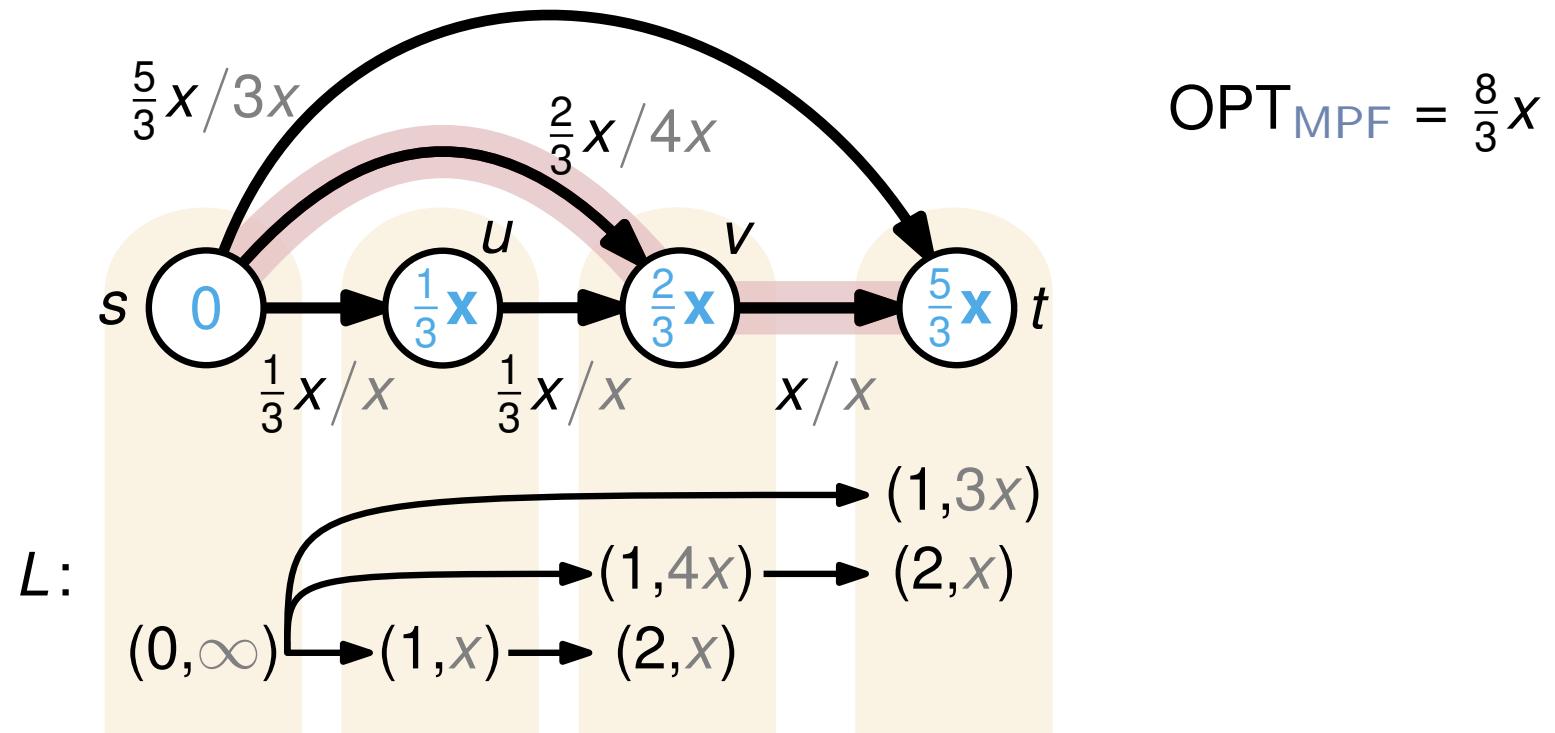
Computing DTP

[Section 5; Grastien et al., 2018]

Description:

- Bicriterial Dijkstra with labels $(\|\pi\|_b, \text{cap}(\pi), V_i)$
- at most $|E|$ labels per vertex

$$b(i, j) := 1 \quad \forall (i, j) \in E$$



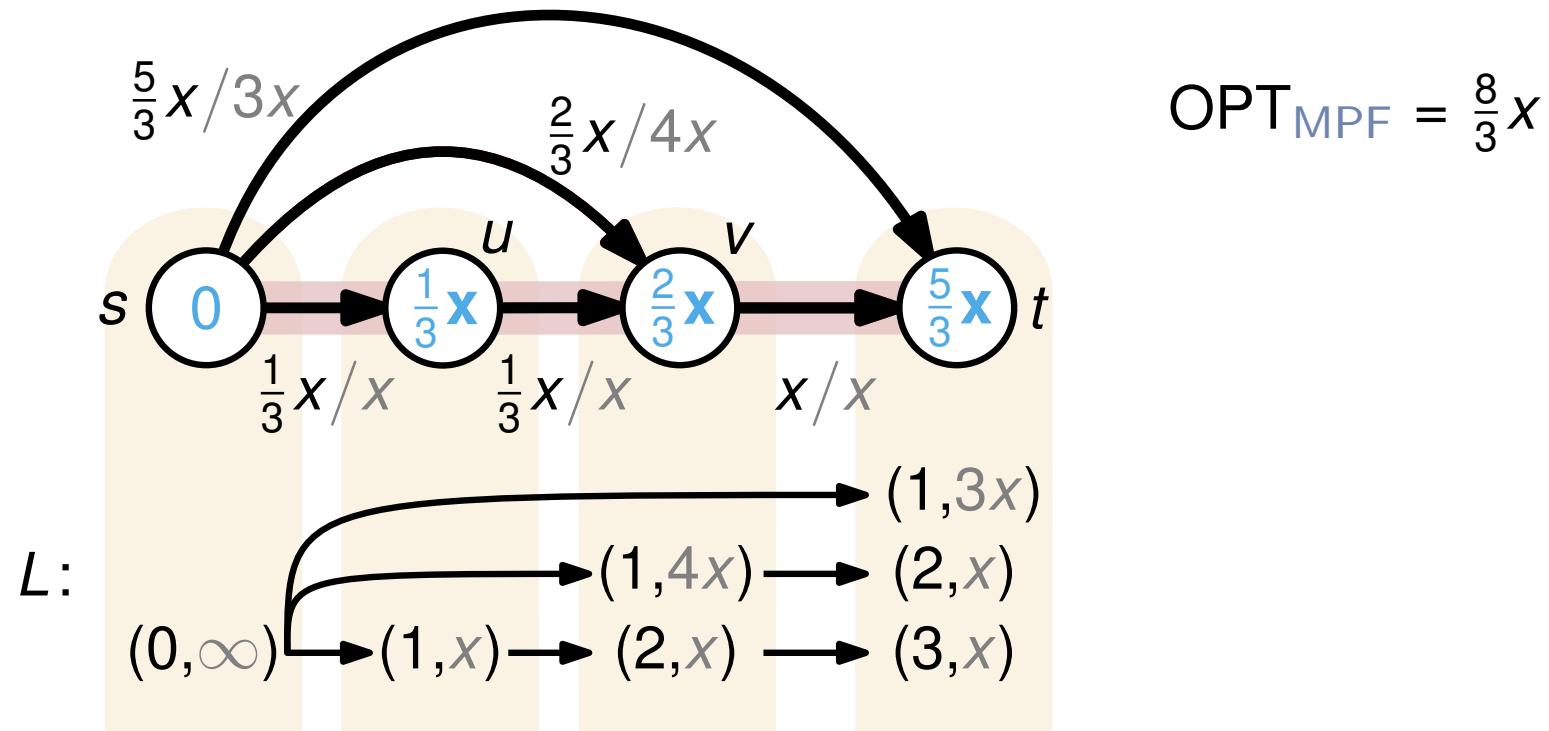
Computing DTP

[Section 5; Grastien et al., 2018]

Description:

- Bicriterial Dijkstra with labels $(\|\pi\|_b, \text{cap}(\pi), V_i)$
- at most $|E|$ labels per vertex

$$b(i, j) := 1 \quad \forall (i, j) \in E$$



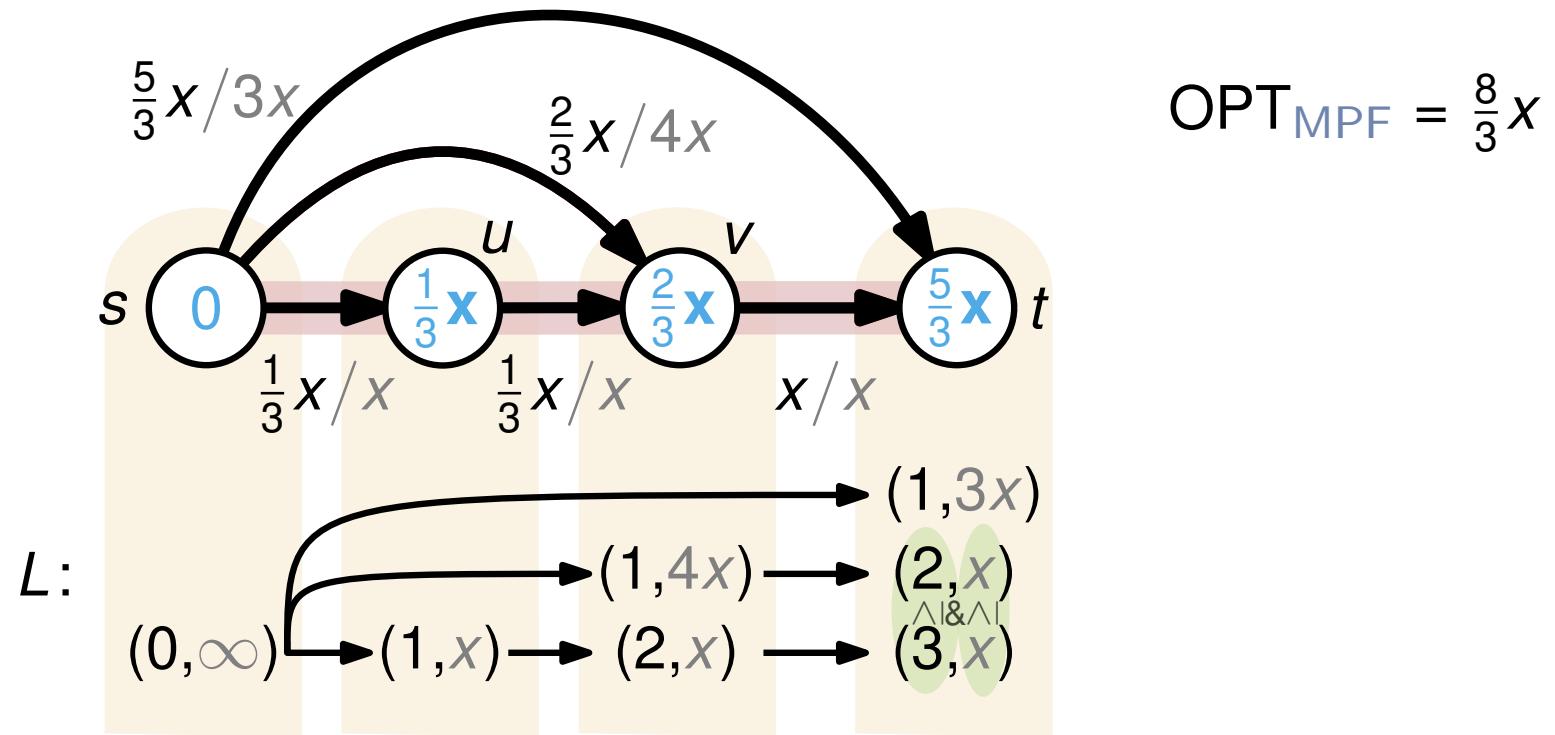
Computing DTP

[Section 5; Grastien et al., 2018]

Description:

- Bicriterial Dijkstra with labels $(\|\pi\|_b, \text{cap}(\pi), V_i)$
- at most $|E|$ labels per vertex

$$b(i, j) := 1 \quad \forall (i, j) \in E$$



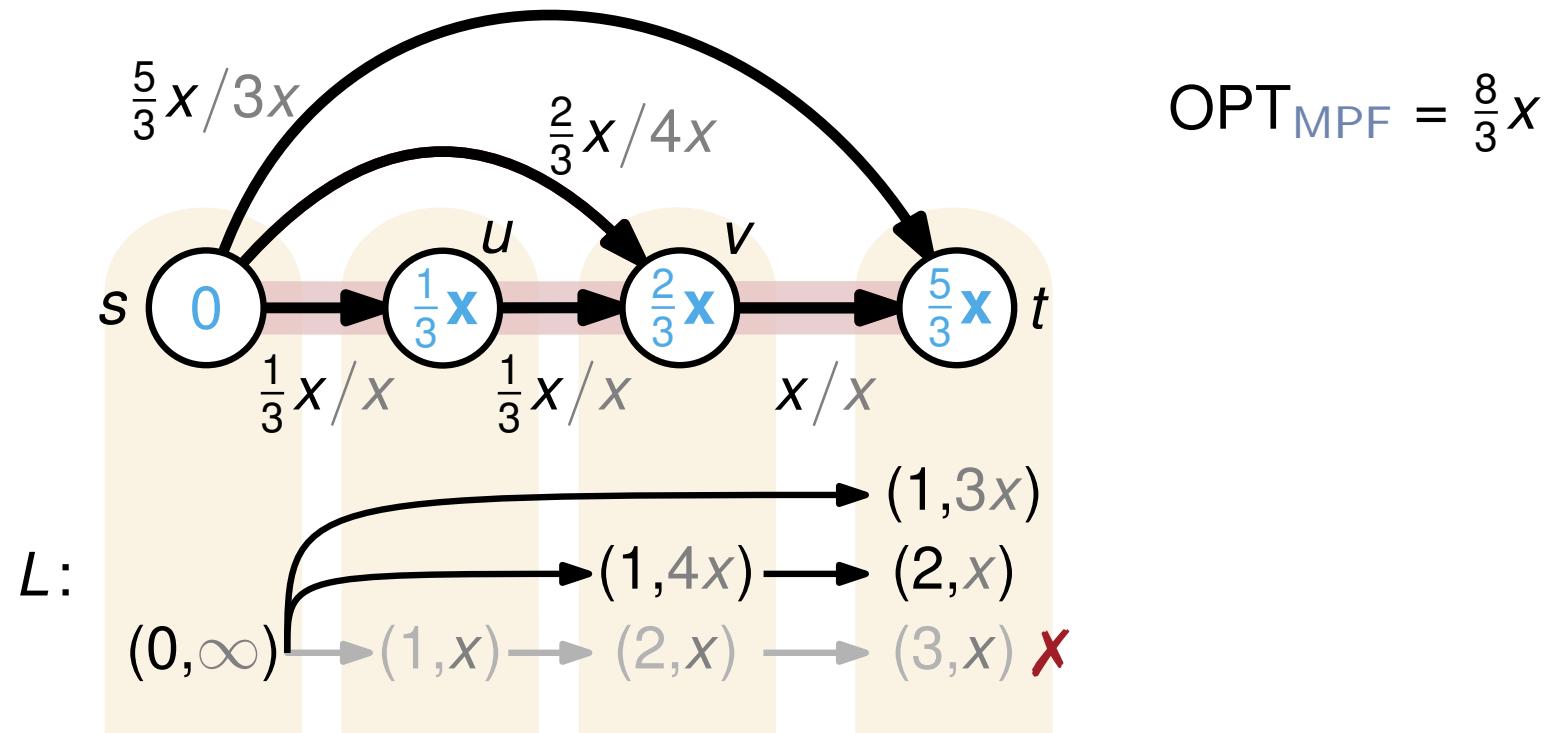
Computing DTP

[Section 5; Grastien et al., 2018]

Description:

- Bicriterial Dijkstra with labels $(\|\pi\|_b, \text{cap}(\pi), V_i)$
- at most $|E|$ labels per vertex

$$b(i, j) := 1 \quad \forall (i, j) \in E$$



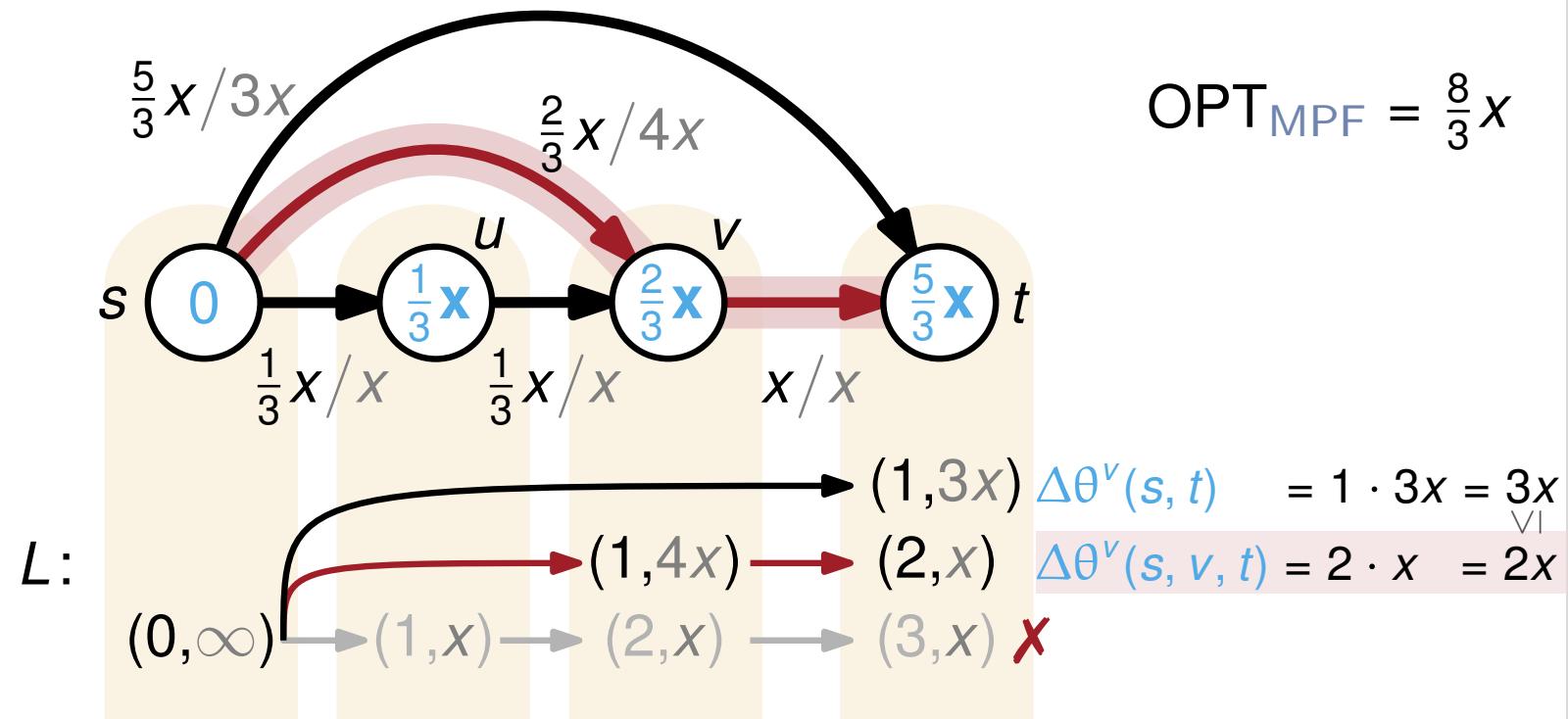
Computing DTP

[Section 5; Grastien et al., 2018]

Description:

- Bicriterial Dijkstra with labels $(\|\pi\|_b, \text{cap}(\pi), V_i)$
- at most $|E|$ labels per vertex

$$b(i, j) := 1 \quad \forall (i, j) \in E$$



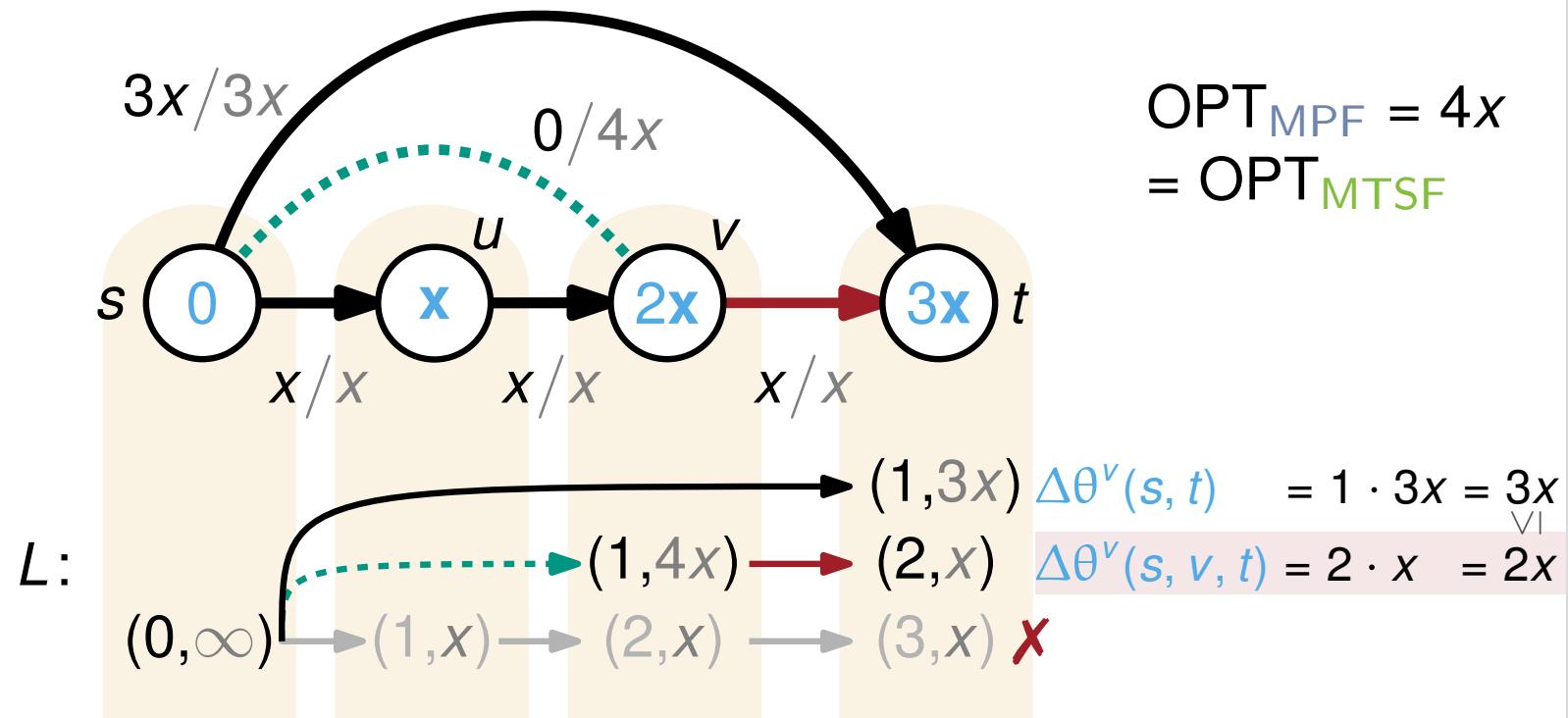
Computing DTP

[Section 5; Grastien et al., 2018]

Description:

- Bicriterial Dijkstra with labels $(\|\pi\|_b, \text{cap}(\pi), V_i)$
- at most $|E|$ labels per vertex

$$b(i, j) := 1 \quad \forall (i, j) \in E$$



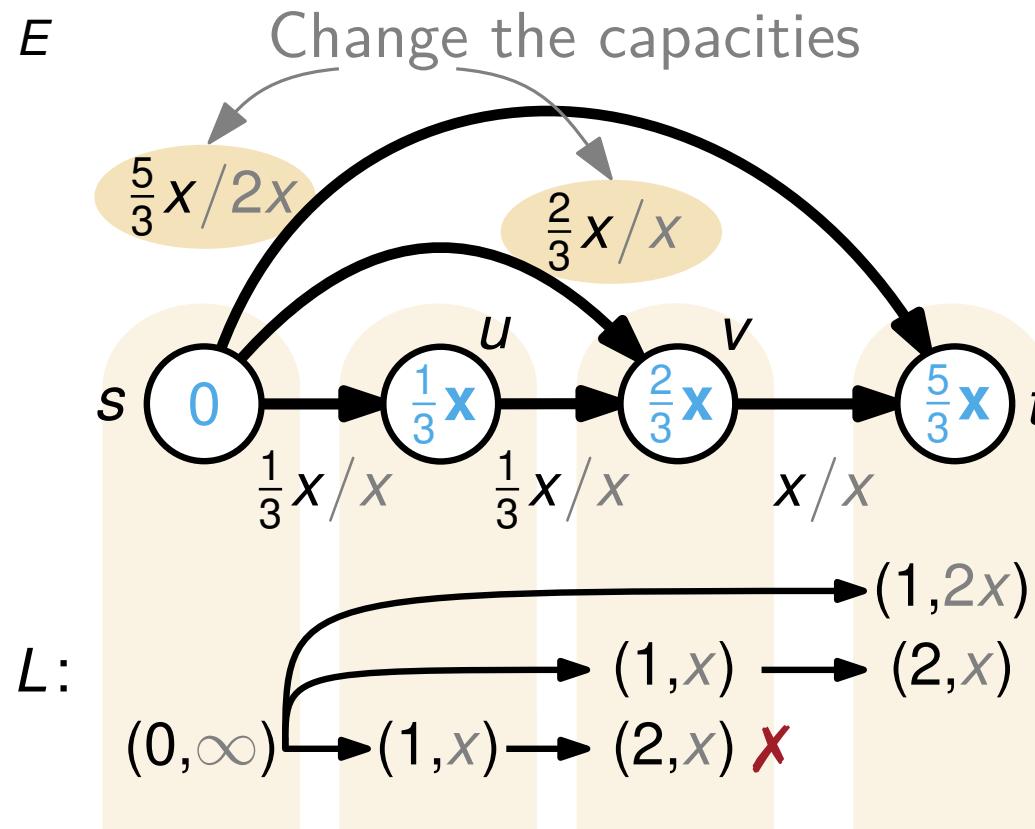
Computing DTP

[Section 5; Grastien et al., 2018]

Description:

- Bicriterial Dijkstra with labels $(\|\pi\|_b, \text{cap}(\pi), V_i)$
- at most $|E|$ labels per vertex

$$b(i, j) := 1 \quad \forall (i, j) \in E$$



$$\text{OPT}_{\text{MPF}} = \frac{8}{3}x$$

Computing DTP

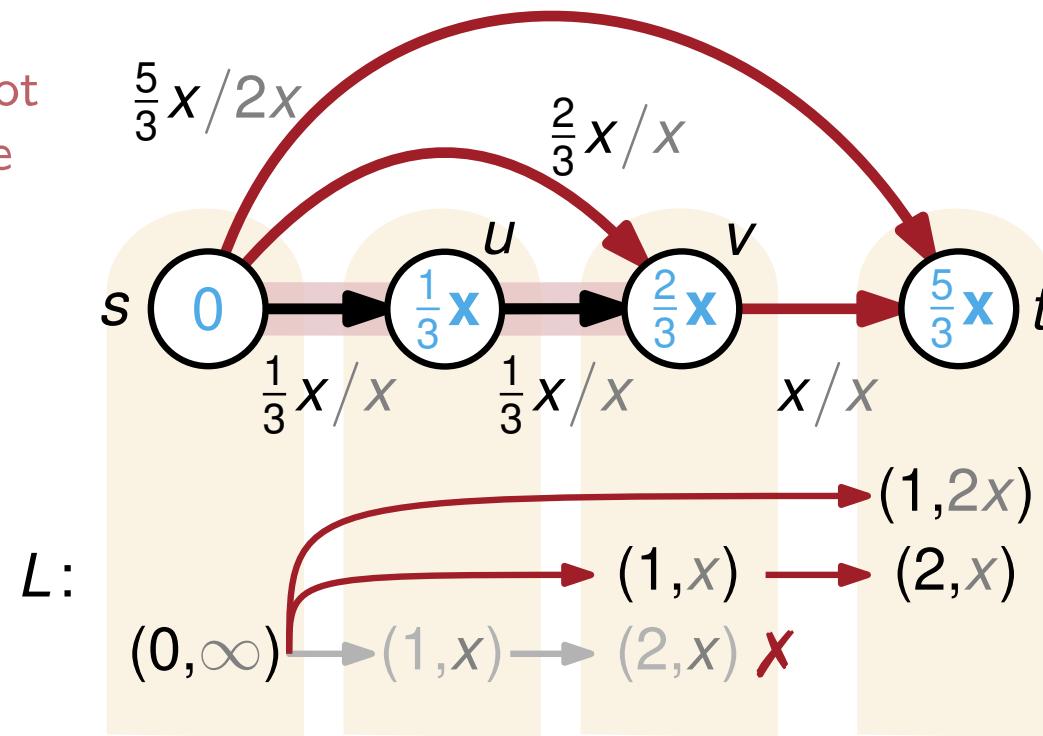
[Section 5; Grastien et al., 2018]

Description:

- Bicriterial Dijkstra with labels $(\|\pi\|_b, \text{cap}(\pi), V_i)$
- at most $|E|$ labels per vertex

$$b(i, j) := 1 \quad \forall (i, j) \in E$$

- DTPs from s do not have to form a tree



Computing DTP

[Section 5; Grastien et al., 2018]

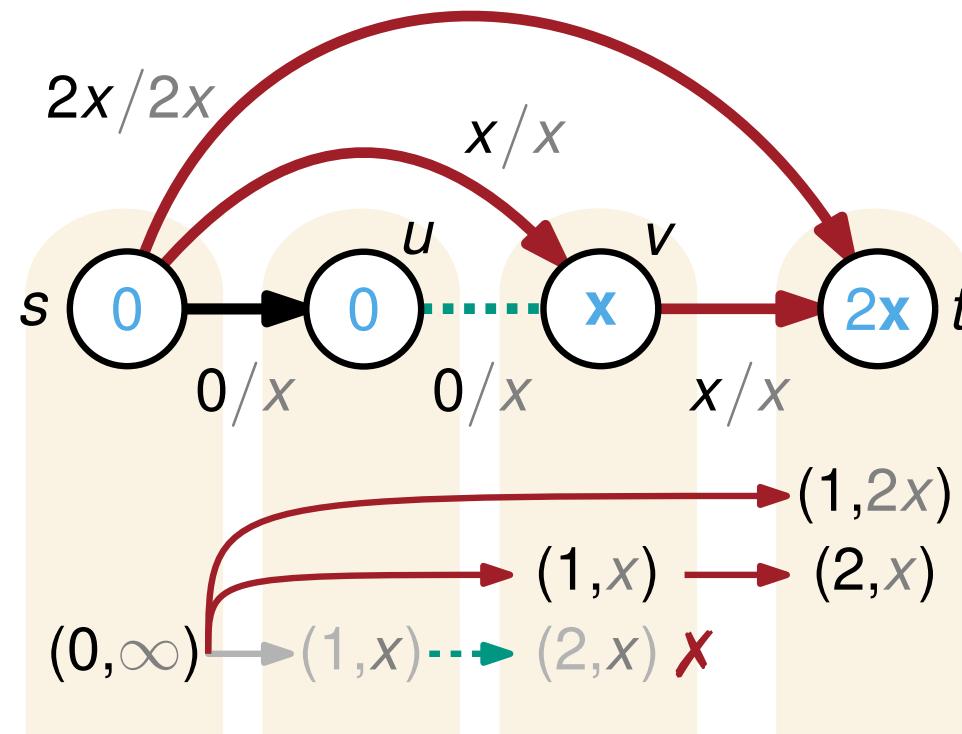
Description:

- Bicriterial Dijkstra with labels $(\|\pi\|_b, \underline{\text{cap}}(\pi), V_i)$
- at most $|E|$ labels per vertex

$$b(i, j) := 1 \quad \forall (i, j) \in E$$

- DTPs from s do not have to form a tree
- Optimal switches do not have to lie on the DTP if the structure is not penrose-minor free

$L:$



$$\begin{aligned} \text{OPT}_{\text{MPF}} &= 3x \\ &= \text{OPT}_{\text{MTSF}} \end{aligned}$$

Running Time Analysis for DTP

[page 345; Grastien et al., 2018]

Data: A network $\mathcal{N} = (G, V_G, V_D, \text{cap}, b)$.	
Result: $\pi_{\text{DTP}}(s, t)$, $\Delta\theta^V(s, t)$, and $D(v)$ with $v \in V$.	
1 $D(u) := L(u) := \emptyset; \forall u \in V;$	\triangleright Initialization
2 $Q := \emptyset;$	
3 $L(s) := \{(0, \infty)\}$	\triangleright Special label for source s
4 $Q.\text{insert}((0, \infty), s, \text{key}((0, \infty)))$;	
5 while $Q \neq \emptyset$ do	\triangleright Visit all vertices
6 (ℓ, u, key) := $Q.\text{deleteMin}()$;	$ V \cdot E $
7 $D(u) := D(u) \cup \{\ell\}$;	$ V \cdot E \cdot \mathcal{O}(\log V)$
8 for $\forall \{u, v\} \in \overleftrightarrow{E}$ do	$ E ^2$
9 $\underline{\text{cap}}(\pi(s, u, v)) := \min(\ell[1], \text{cap}(u, v));$	$ E ^2 \cdot \mathcal{O}(1)$
10 $\ell_{\text{new}}(v) := \left(\ell[0] + \frac{1}{b(u, v)}, \underline{\text{cap}}(\pi(s, u, v))\right);$	$ E ^2 \cdot \mathcal{O}(1)$
11 If $\text{isReachable}(V \setminus \{v\}, \ell, s)$ then	$ E ^2 \cdot \mathcal{O}(2^{ V } V \cdot E)$
12 If $\ell_{\text{new}}(v) \in L(v)$ then	$ E ^2 \cdot \mathcal{O}(1)$
13 $\text{parent}(\ell_{\text{new}}(v)) := \text{parent}(\ell_{\text{new}}(v)) \cup \{\ell\};$	$ E ^2 \cdot \mathcal{O}(1)$
14 else if not $L(v)$ dominates $\ell_{\text{new}}(v)$ then	
15 $L(v).\text{deleteDominatedLabels}(\ell_{\text{new}}(v));$	$ E ^2 \cdot \mathcal{O}(E)$
16 $Q.\text{deleteDominatedLabels}(\ell_{\text{new}}(v), v);$	$ E ^2 \cdot \mathcal{O}(E)$
17 $L(v).\text{insert}(\ell_{\text{new}}(v));$	$ E ^2 \cdot \mathcal{O}(1)$
18 $Q.\text{insert}(\ell_{\text{new}}(v), v, \text{key}(\ell_{\text{new}}(v)))$;	$ E ^2 \cdot \mathcal{O}(1)$
19 $\text{parent}(\ell_{\text{new}}(v)) := \{\ell\};$	$ E ^2 \cdot \mathcal{O}(1)$
20 end	iterate over all labels in the bag, $ E $ tests per bag
21 end	
22 end	
23 end	
24 return $\left(\begin{array}{l} \pi_{\text{DTP}}(s, t) := \text{getPaths}(s, t), \triangleright \text{Build paths from parent} \\ \underline{\Delta\theta^V}(s, t) := \min_{\ell \in D(t)} \{\ell[0] \cdot \ell[1]\}, \\ D(\cdot) \end{array} \right)$	Königsberg Bridge Problem $\sum_{v \in V} \deg(v) = 2 E $ Fibonacci-heap Q insert $\mathcal{O}(1)$ decreaseKey $\mathcal{O}(1)$ amortized delMin $\mathcal{O}(\log V)$ amortized

Running Time Analysis for DTP

[page 345; Grastien et al., 2018]

Data: A network $\mathcal{N} = (G, V_G, V_D, \text{cap}, b)$.

Result: $\pi_{\text{DTP}}(s, t)$, $\Delta\theta^V(s, t)$, and $D(v)$ with $v \in V$.

```

1  $D(u) := L(u) := \emptyset; \forall u \in V;$                                 ▷ Initialization
2  $Q := \emptyset;$ 
3  $L(s) := \{(0, \infty)\}$                                               ▷ Special label for source  $s$ 
4  $Q.\text{insert}((0, \infty), s, \text{key}((0, \infty)))$ ;
5 while  $Q \neq \emptyset$  do
6    $(\ell, u, \text{key}) := Q.\text{deleteMin}();$ 
7    $D(u) := D(u) \cup \{\ell\};$                                          ▷ Visit all vertices
8   for  $\forall \{u, v\} \in \overleftrightarrow{E}$  do                                     ▷ Check adjacent vertices

```

$$\left. \begin{array}{l} \\ \\ \\ \end{array} \right\} \mathcal{O}(|V|)$$

$$\begin{array}{l} |V| \cdot |E| \\ |V| \cdot |E| \cdot \mathcal{O}(\log |V|) \end{array}$$

$$|E|^2$$

On general graphs the DTP algorithm runs in time $\mathcal{O}(2^{|V|}|V| \cdot |E|^3)$.

```

13   if  $\ell_{\text{new}}(v) \in L(v)$  then                                         | $E|^2 \cdot \mathcal{O}(1)$ 
14      $\text{parent}(\ell_{\text{new}}(v)) := \text{parent}(\ell_{\text{new}}(v)) \cup \{\ell\};$ 
15   else if not  $L(v)$  dominates  $\ell_{\text{new}}(v)$  then                         | $E|^2 \cdot \mathcal{O}(|E|)$ 
16      $L(v).\text{deleteDominatedLabels}(\ell_{\text{new}}(v));$ 
17      $Q.\text{deleteDominatedLabels}(\ell_{\text{new}}(v), v);$                            | $E|^2 \cdot \mathcal{O}(|E|)$ 
18      $L(v).\text{insert}(\ell_{\text{new}}(v));$                                          | $E|^2 \cdot \mathcal{O}(1)$ 
19      $Q.\text{insert}(\ell_{\text{new}}(v), v, \text{key}(\ell_{\text{new}}(v)));$                       | $E|^2 \cdot \mathcal{O}(1)$ 
20      $\text{parent}(\ell_{\text{new}}(v)) := \{\ell\};$                                          | $E|^2 \cdot \mathcal{O}(1)$ 
21   end                                                               → iterate over all labels in the bag,  $|E|$  tests per bag
22   end                                                               Fibonacci-heap  $Q$ 
23   end                                                               insert  $\mathcal{O}(1)$ 
24 return  $\left( \begin{array}{l} \pi_{\text{DTP}}(s, t) := \text{getPaths}(s, t), \triangleright \text{Build paths from parent} \\ \Delta\theta^V(s, t) := \min_{\ell \in D(t)} \{\ell[0] \cdot \ell[1]\}, \\ D(\cdot) \end{array} \right)$ 
25                                                               decreaseKey  $\mathcal{O}(1)$ 
26                                                               delMin  $\mathcal{O}(\log |V|)$  amortized

```

Running Time Analysis for DTP

Data: A network $\mathcal{N} = (G, V_G, V_D, \text{cap}, b)$.

Result: $\pi_{\text{DTP}}(s, t)$, $\Delta\theta^v(s, t)$, and $D(v)$ with $v \in V$.

$$^1 D(u) := L(u) := \emptyset; \forall u \in V;$$

$$^2\mid Q:=\emptyset;$$

$$^3 L(s) := \{(0, \infty)\}$$

```
4 Q.insert((0, infinity), s, key((0, infinity)));
```

$$\mathcal{O}(|V|)$$

5 while $Q \neq \emptyset$ do

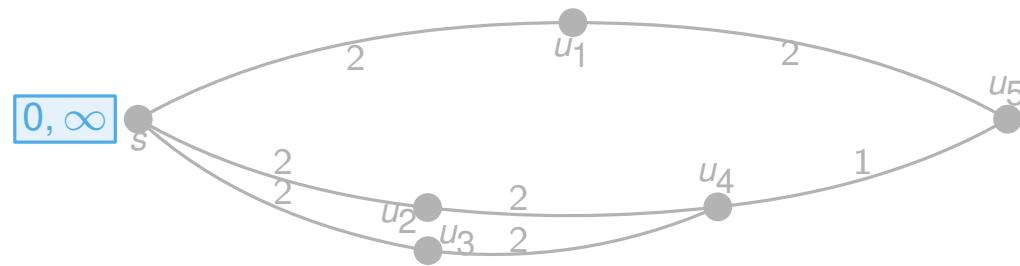
6 | $(\ell, u, \text{key}) := Q.\text{deleteMin}();$

$$7 \qquad D(u) := D(u) \cup \{\ell\};$$

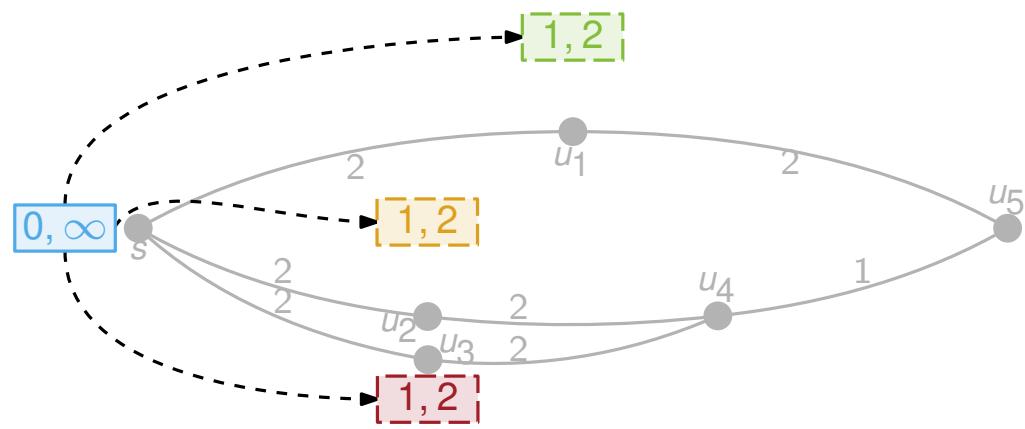
for $\forall \{u, v\} \in \overleftrightarrow{E}$ do

On general graphs a **DTP** algorithm exists that runs in polynomial time and calculates one **DTP** between each pair of u and v .

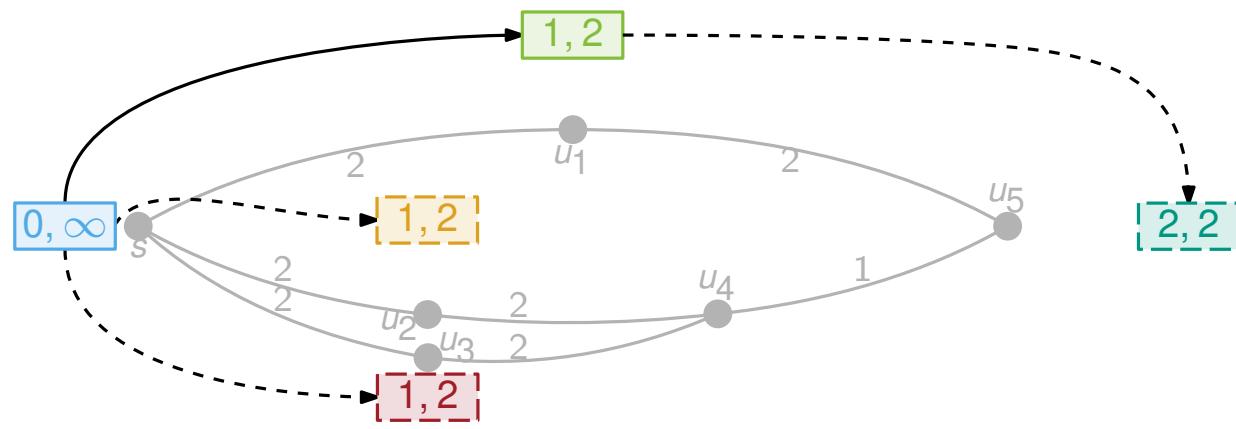
isReachable [page 354; Grastien et al., 2018]



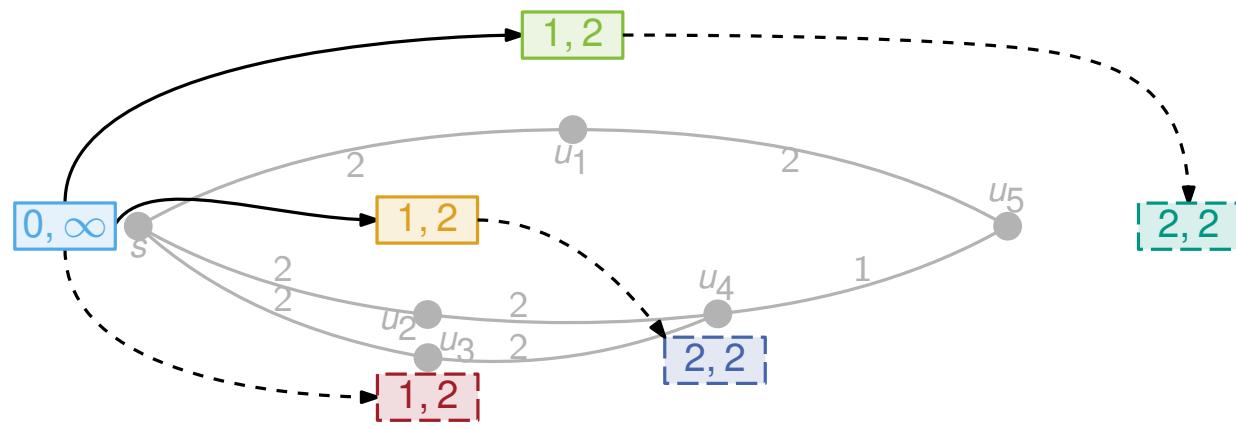
isReachable [page 354; Grastien et al., 2018]



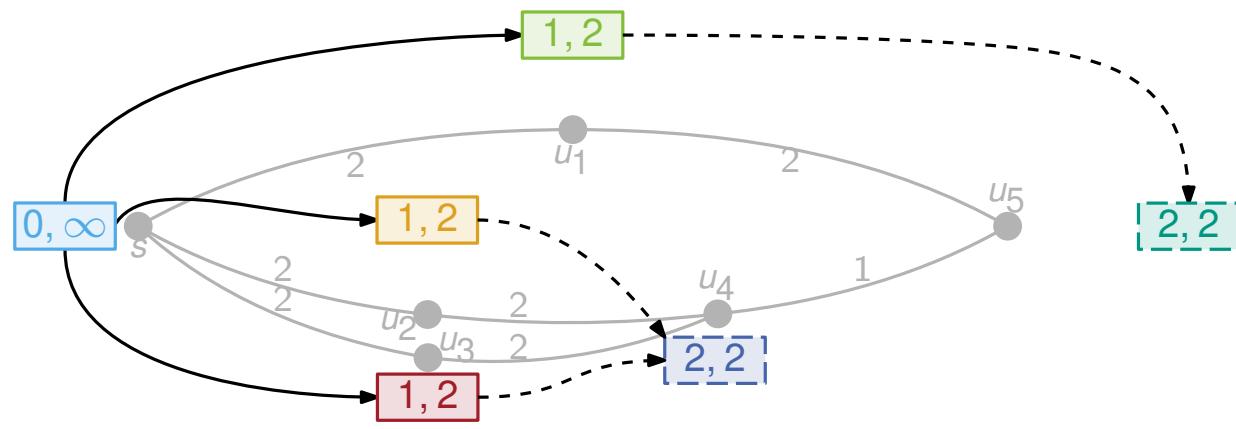
isReachable [page 354; Grastien et al., 2018]



isReachable [page 354; Grastien et al., 2018]

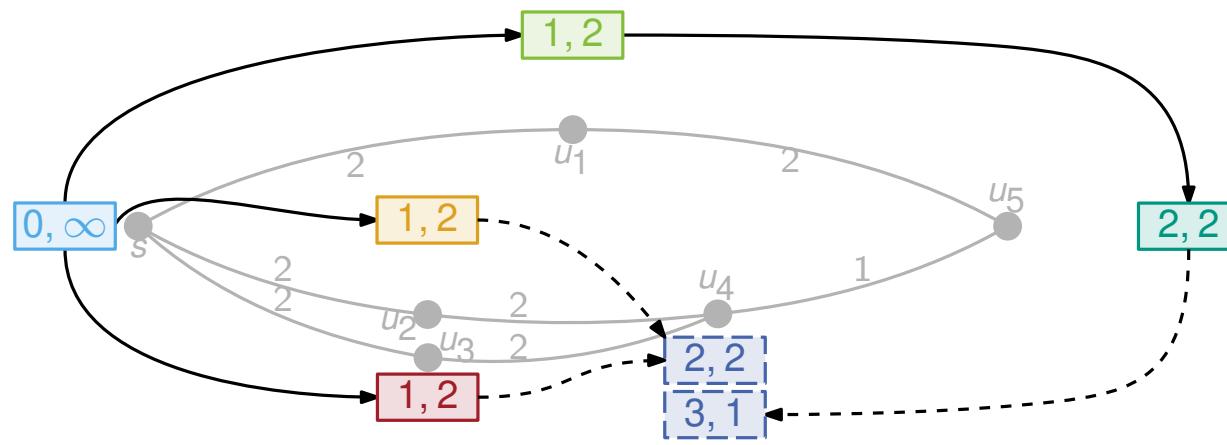


isReachable [page 354; Grastien et al., 2018]



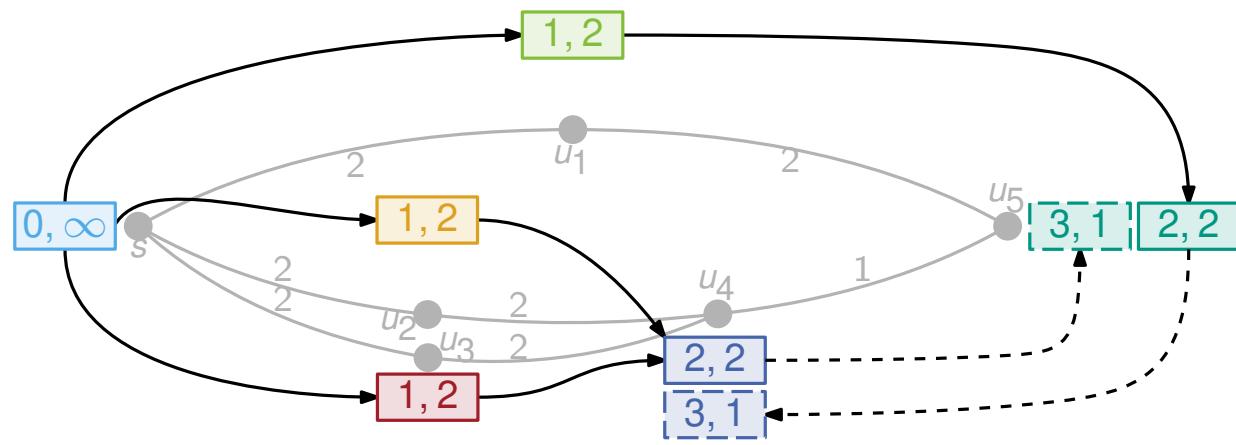
isReachable

[page 354; Grastien et al., 2018]



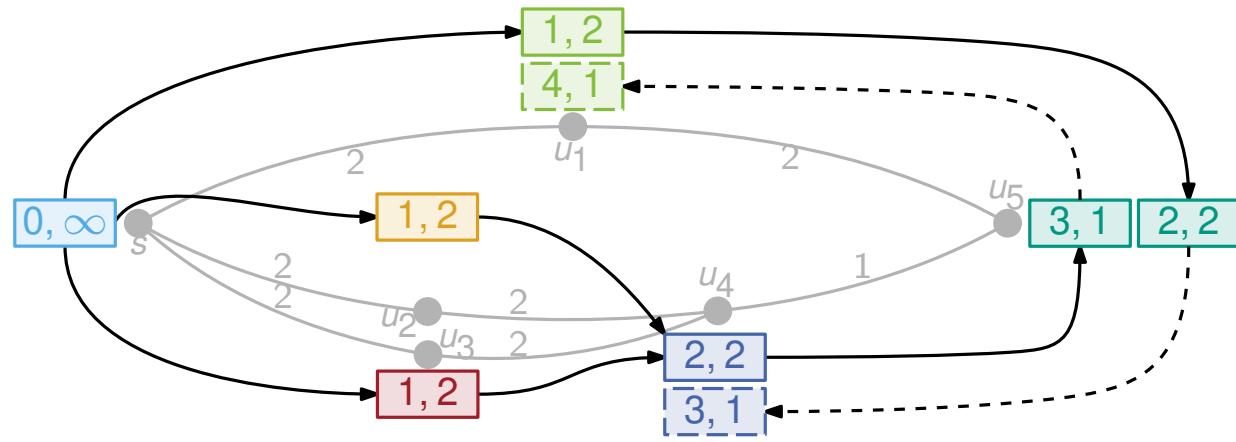
isReachable

[page 354; Grastien et al., 2018]



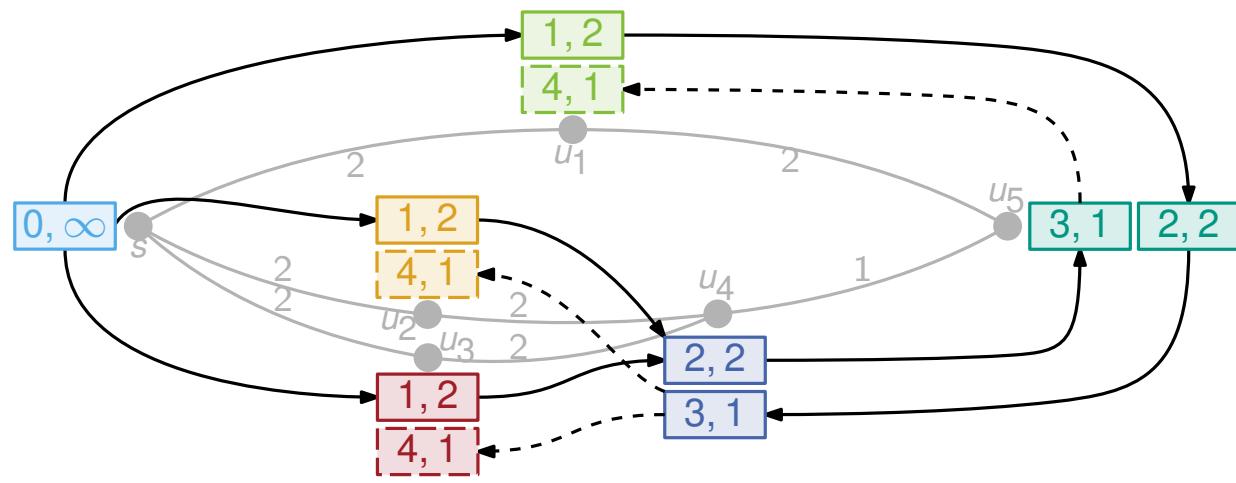
isReachable

[page 354; Grastien et al., 2018]



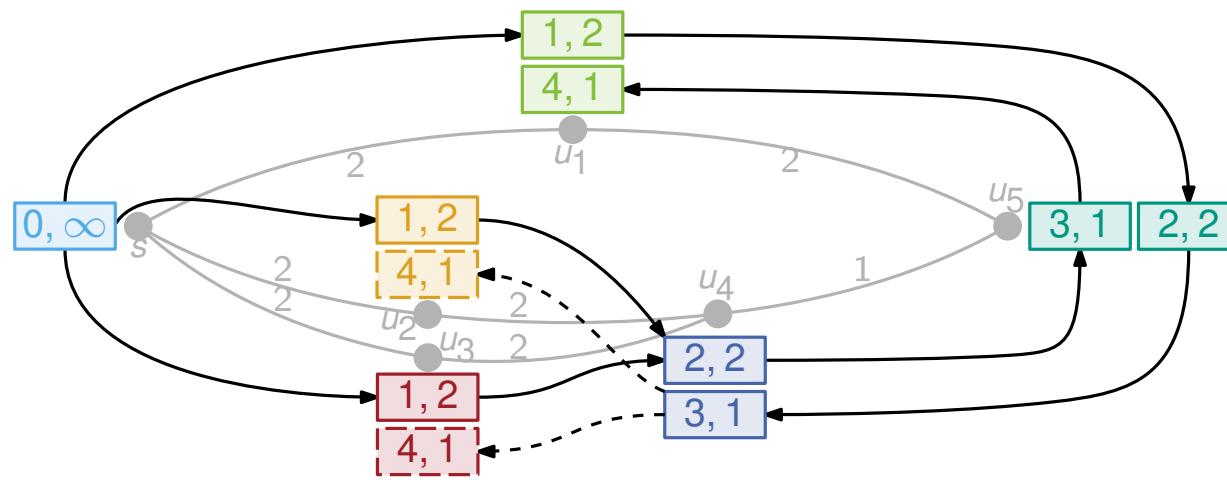
isReachable

[page 354; Grastien et al., 2018]



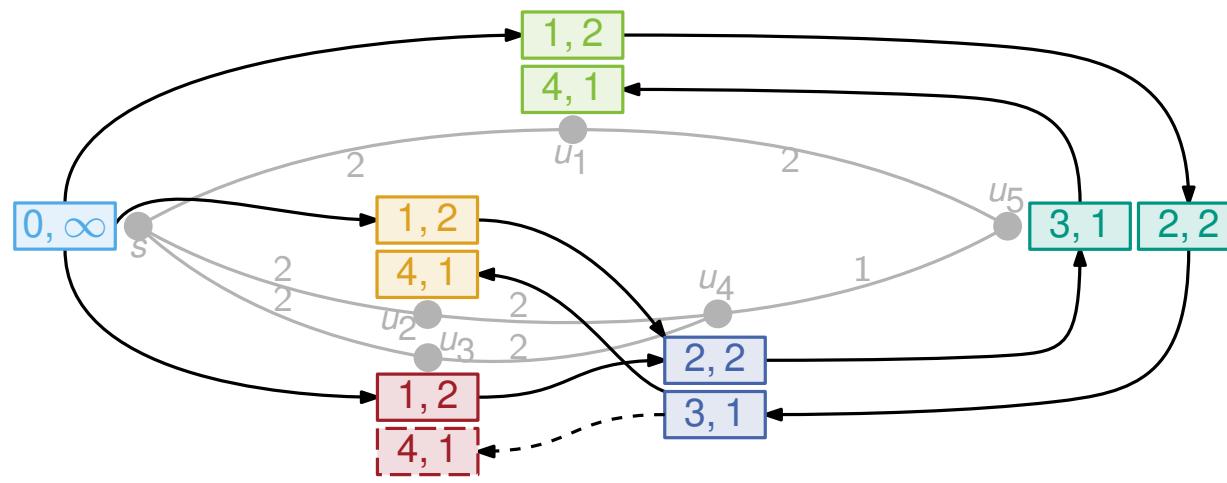
isReachable

[page 354; Grastien et al., 2018]



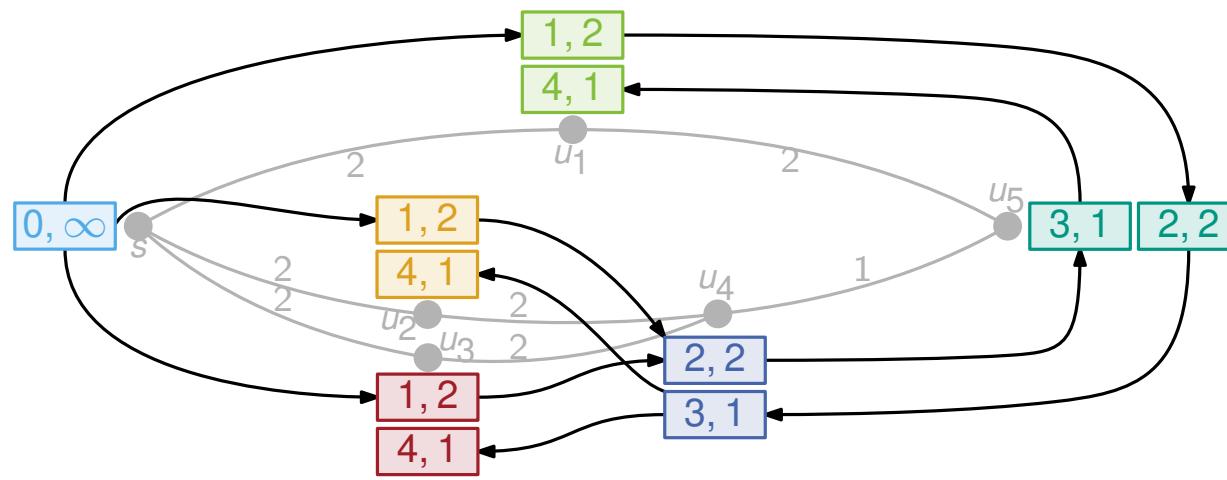
isReachable

[page 354; Grastien et al., 2018]



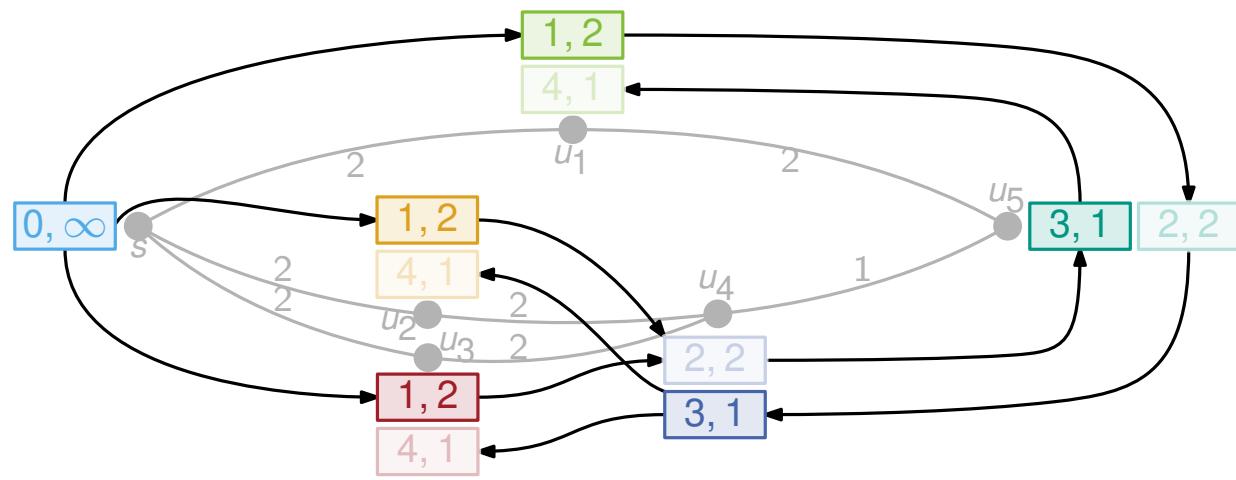
isReachable

[page 354; Grastien et al., 2018]



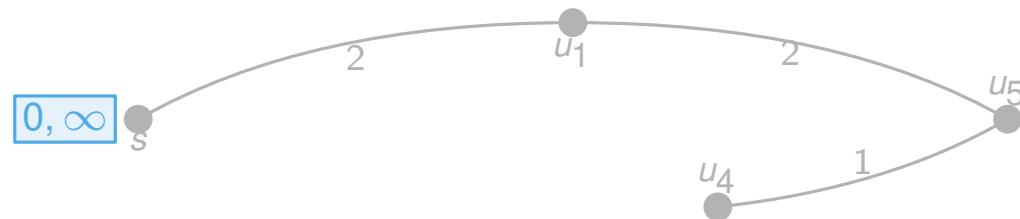
isReachable

[page 354; Grastien et al., 2018]



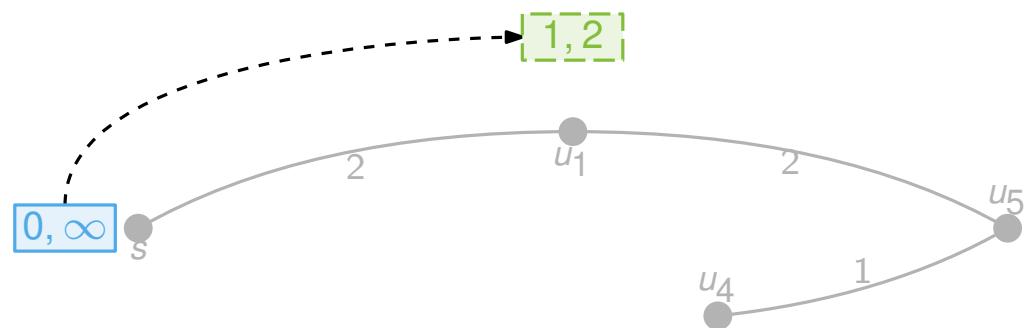
isReachable [page 354; Grastien et al., 2018]

- Does it always work like this?



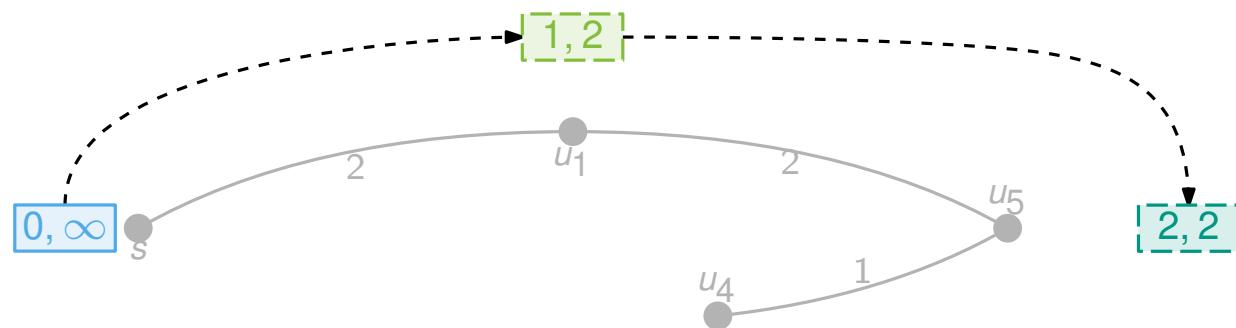
isReachable [page 354; Grastien et al., 2018]

- Does it always work like this?



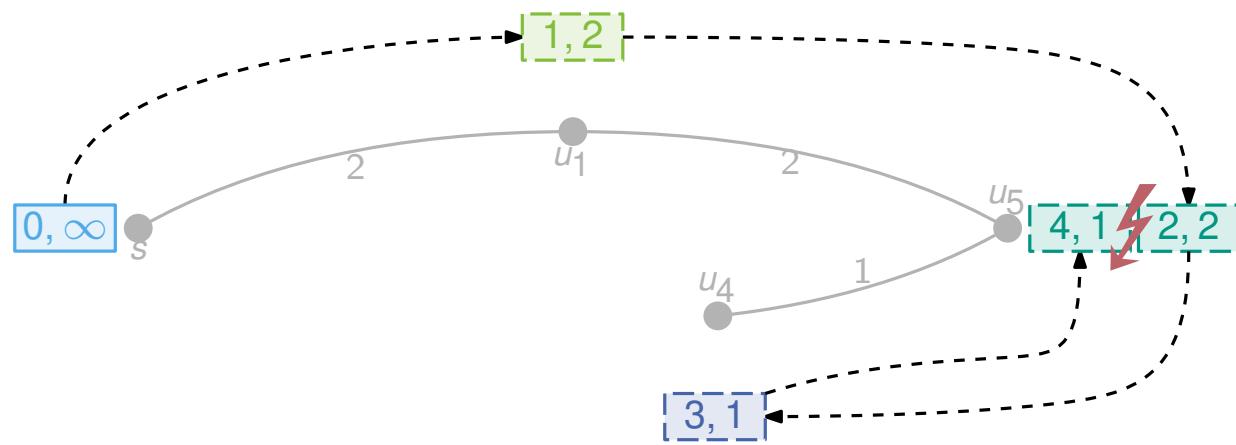
isReachable [page 354; Grastien et al., 2018]

- Does it always work like this?



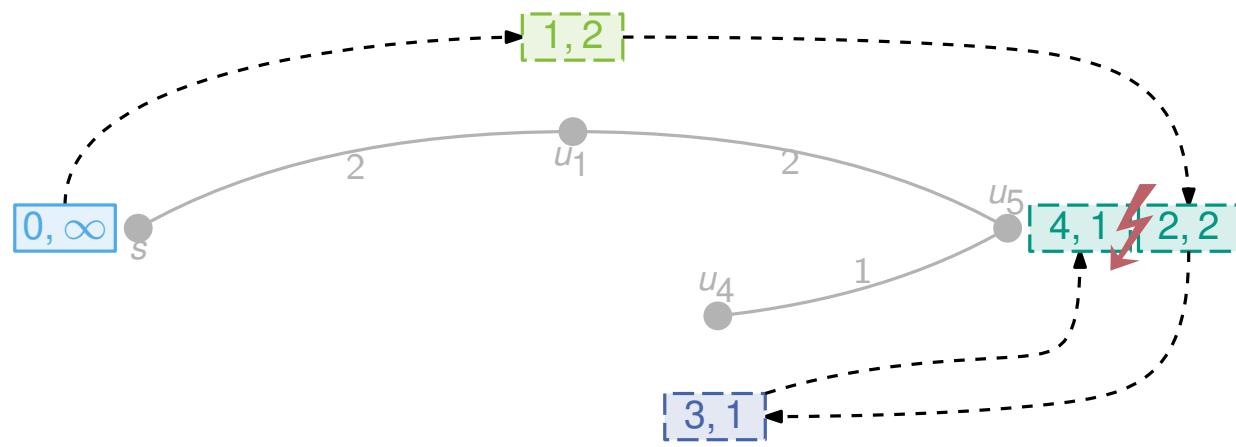
isReachable [page 354; Grastien et al., 2018]

- Does it always work like this?



isReachable [page 354; Grastien et al., 2018]

- Does it always work like this?



⇒ Check if label corresponds to simple path

Rainbow Paths – FPT Algorithm

[Theorem 11; Uchizawa et al., 2013]

- FPT algorithm w.r.t. the number of colors

Rainbow $s-t$ -Path ($s - t - \text{RP}$)

Instance: A directed acyclic graph $G = (V, E)$, a coloring $c: V \rightarrow \mathbb{N}$, and $s, t \in V$.

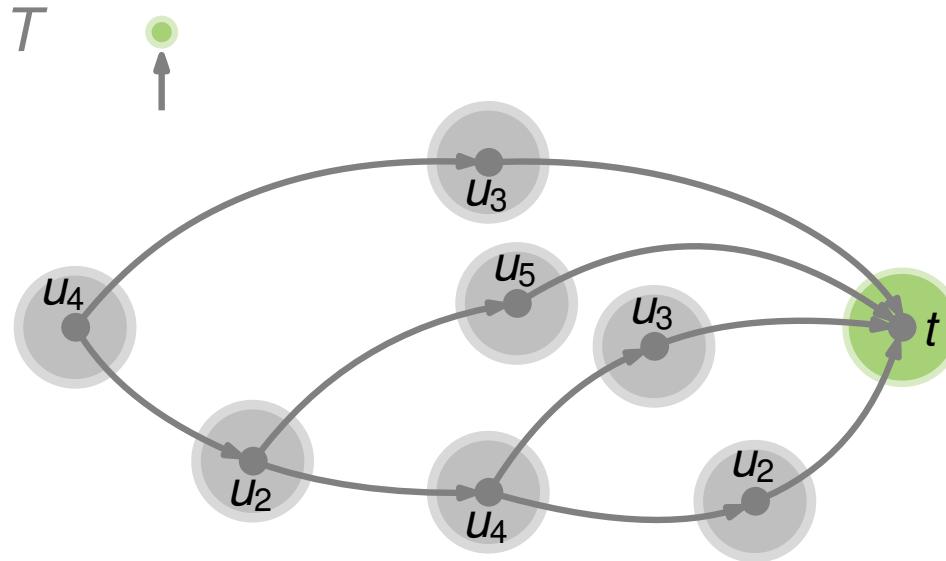
Question: Is there an $s-t$ -path π in G such that all vertices of π have different colors?

But: n colors \Rightarrow not a polynomial time algorithm(?)

Rainbow Paths – FPT Algorithm

[Theorem 11; Uchizawa et al., 2013]

- FPT algorithm w.r.t. the number of colors

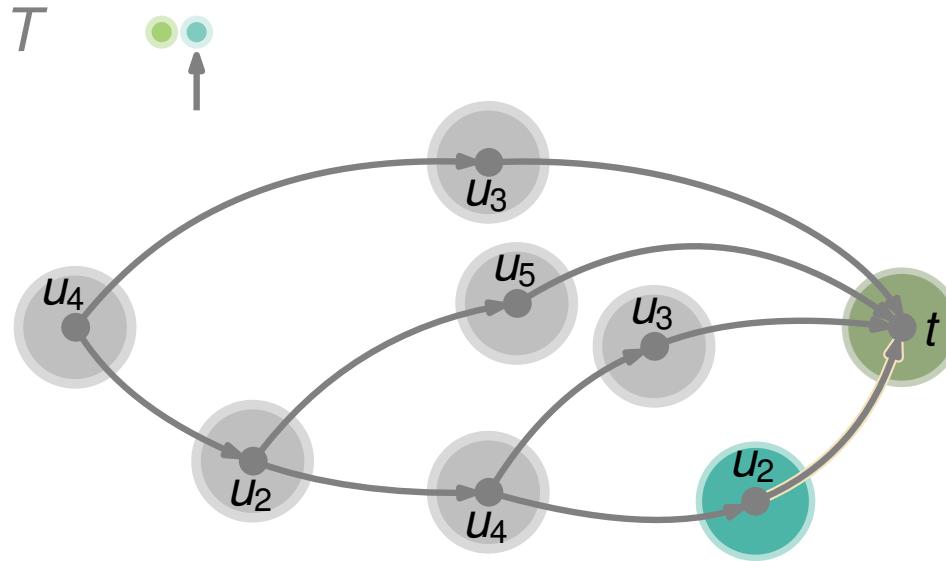


But: n colors \Rightarrow not a polynomial time algorithm(?)

Rainbow Paths – FPT Algorithm

[Theorem 11; Uchizawa et al., 2013]

- FPT algorithm w.r.t. the number of colors

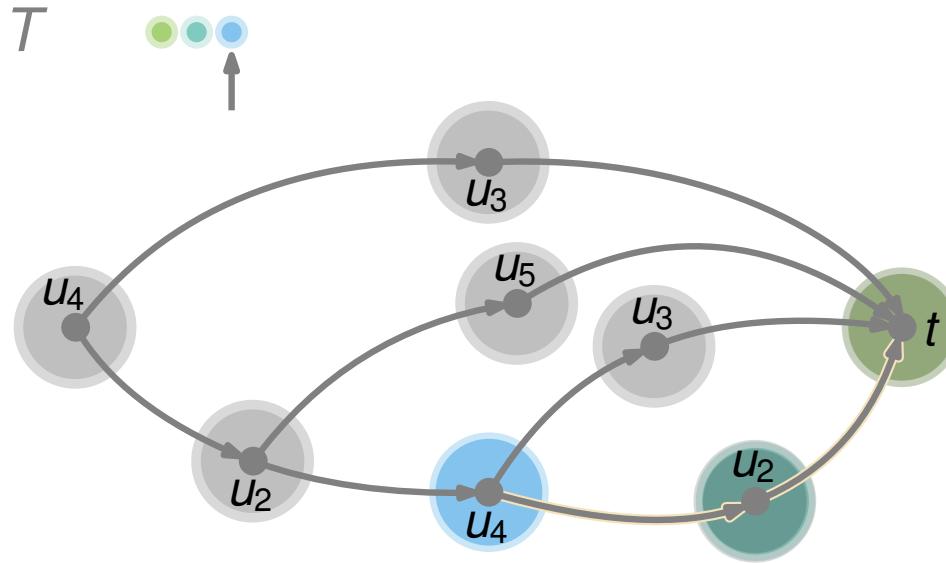


But: n colors \Rightarrow not a polynomial time algorithm(?)

Rainbow Paths – FPT Algorithm

[Theorem 11; Uchizawa et al., 2013]

- FPT algorithm w.r.t. the number of colors

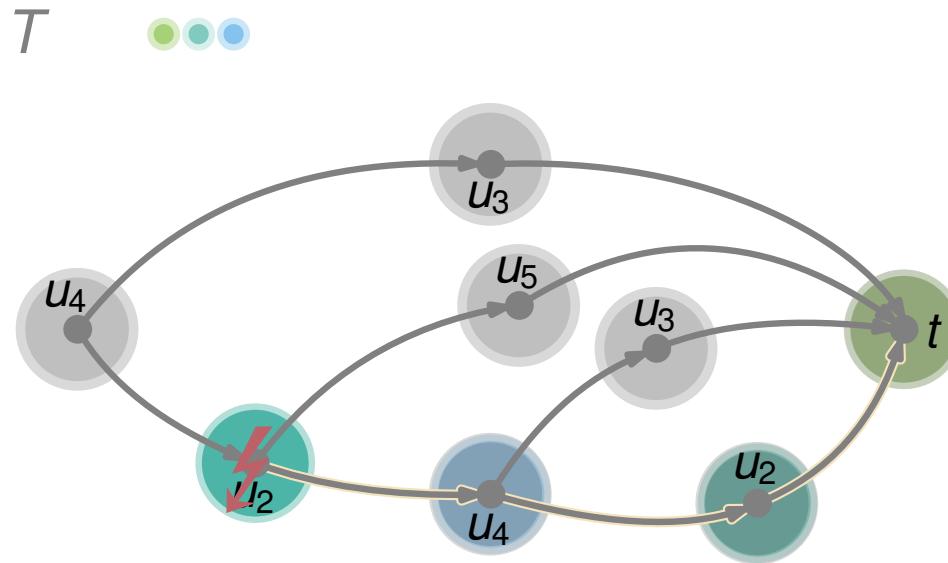


But: n colors \Rightarrow not a polynomial time algorithm(?)

Rainbow Paths – FPT Algorithm

[Theorem 11; Uchizawa et al., 2013]

- FPT algorithm w.r.t. the number of colors

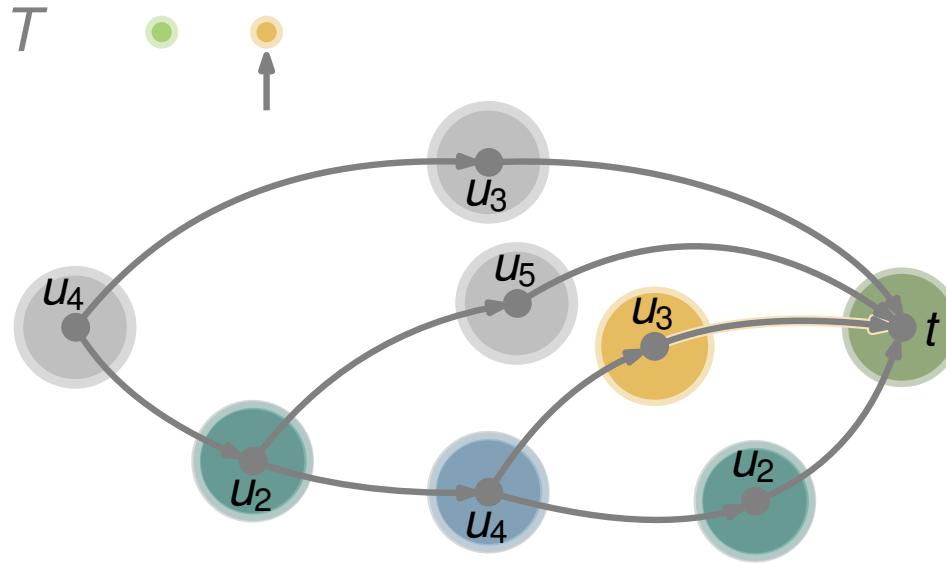


But: n colors \Rightarrow not a polynomial time algorithm(?)

Rainbow Paths – FPT Algorithm

[Theorem 11; Uchizawa et al., 2013]

- FPT algorithm w.r.t. the number of colors

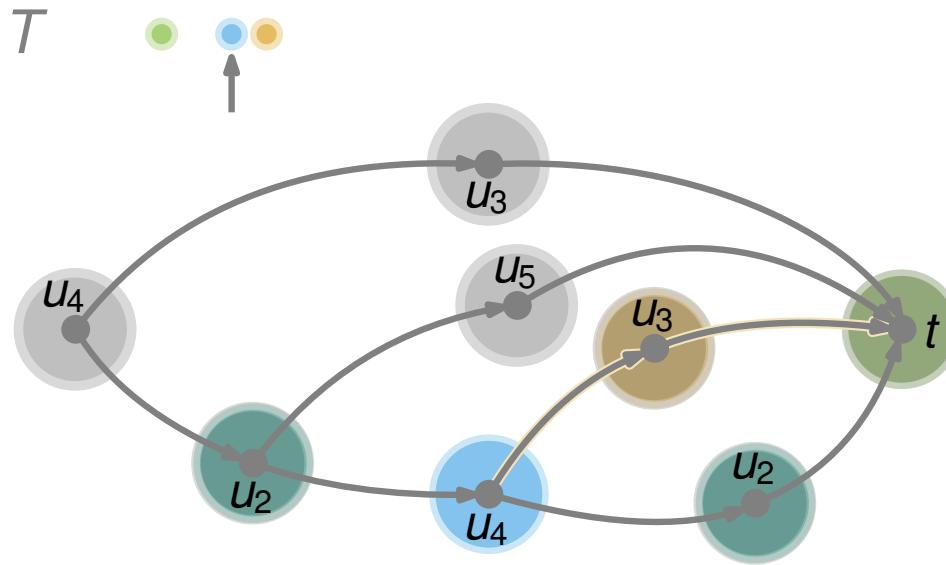


But: n colors \Rightarrow not a polynomial time algorithm(?)

Rainbow Paths – FPT Algorithm

[Theorem 11; Uchizawa et al., 2013]

- FPT algorithm w.r.t. the number of colors

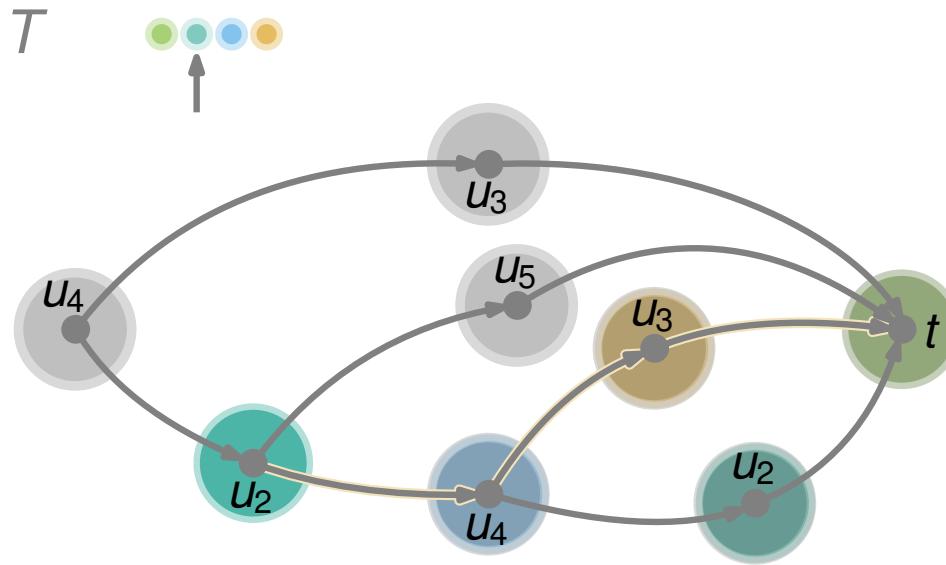


But: n colors \Rightarrow not a polynomial time algorithm(?)

Rainbow Paths – FPT Algorithm

[Theorem 11; Uchizawa et al., 2013]

- FPT algorithm w.r.t. the number of colors

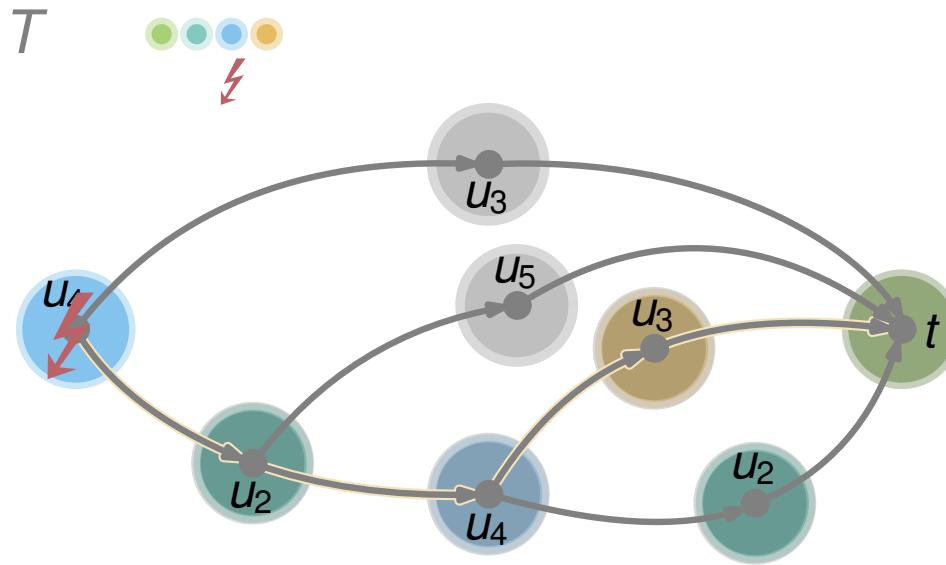


But: n colors \Rightarrow not a polynomial time algorithm(?)

Rainbow Paths – FPT Algorithm

[Theorem 11; Uchizawa et al., 2013]

- FPT algorithm w.r.t. the number of colors

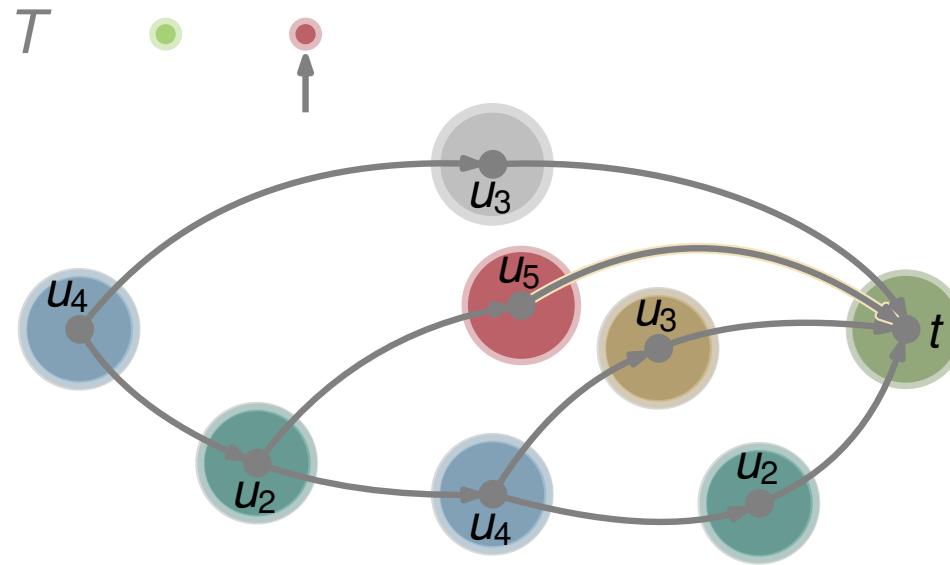


But: n colors \Rightarrow not a polynomial time algorithm(?)

Rainbow Paths – FPT Algorithm

[Theorem 11; Uchizawa et al., 2013]

- FPT algorithm w.r.t. the number of colors

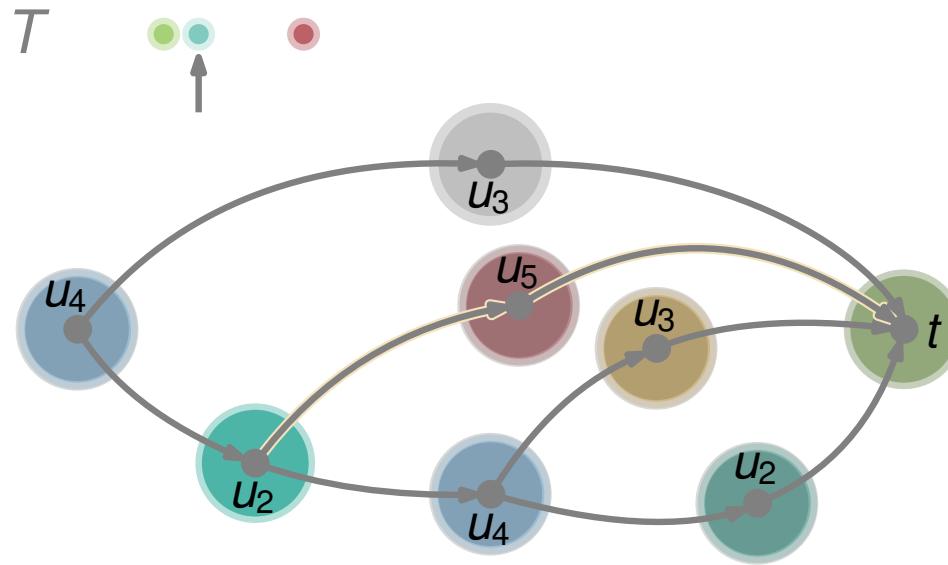


But: n colors \Rightarrow not a polynomial time algorithm(?)

Rainbow Paths – FPT Algorithm

[Theorem 11; Uchizawa et al., 2013]

- FPT algorithm w.r.t. the number of colors

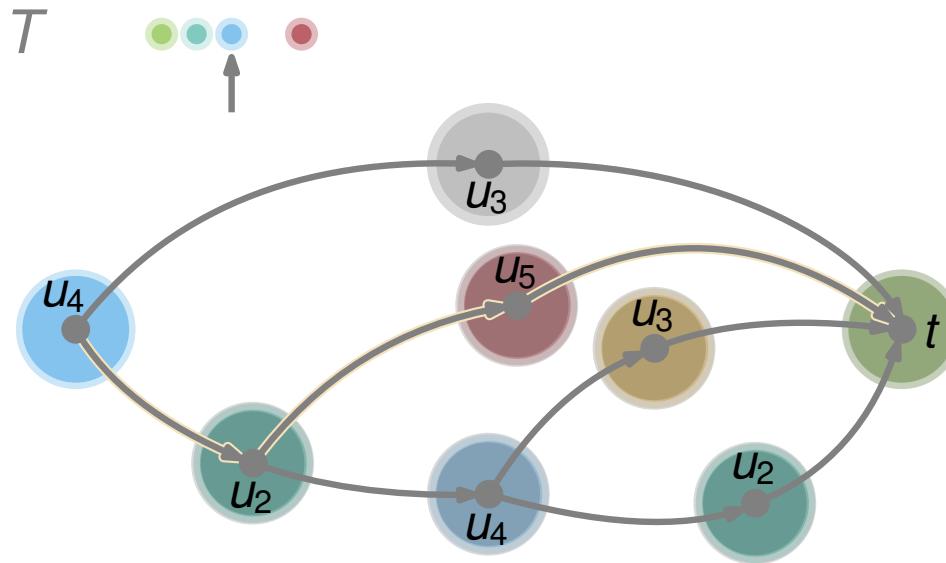


But: n colors \Rightarrow not a polynomial time algorithm(?)

Rainbow Paths – FPT Algorithm

[Theorem 11; Uchizawa et al., 2013]

- FPT algorithm w.r.t. the number of colors

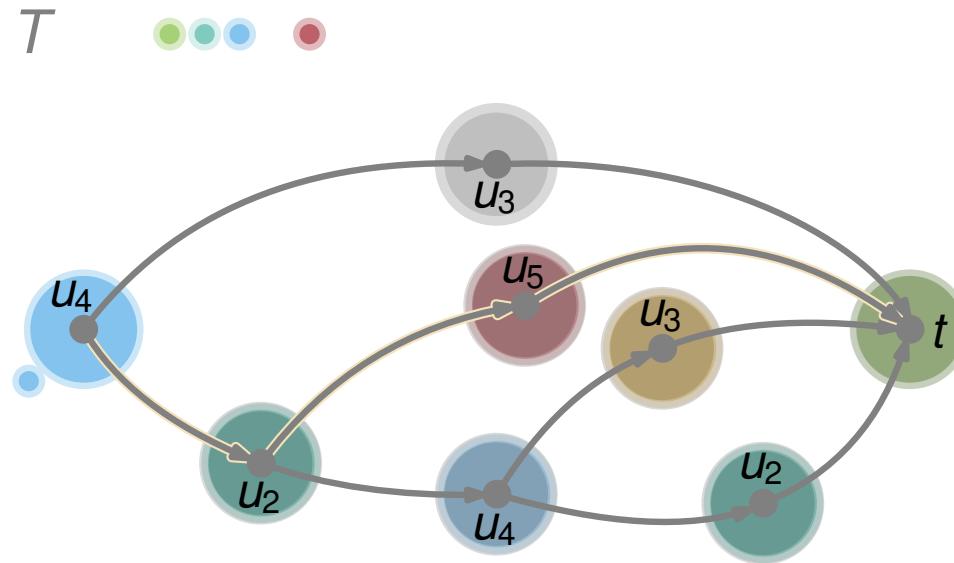


But: n colors \Rightarrow not a polynomial time algorithm(?)

Rainbow Paths – FPT Algorithm

[Theorem 11; Uchizawa et al., 2013]

- FPT algorithm w.r.t. the number of colors

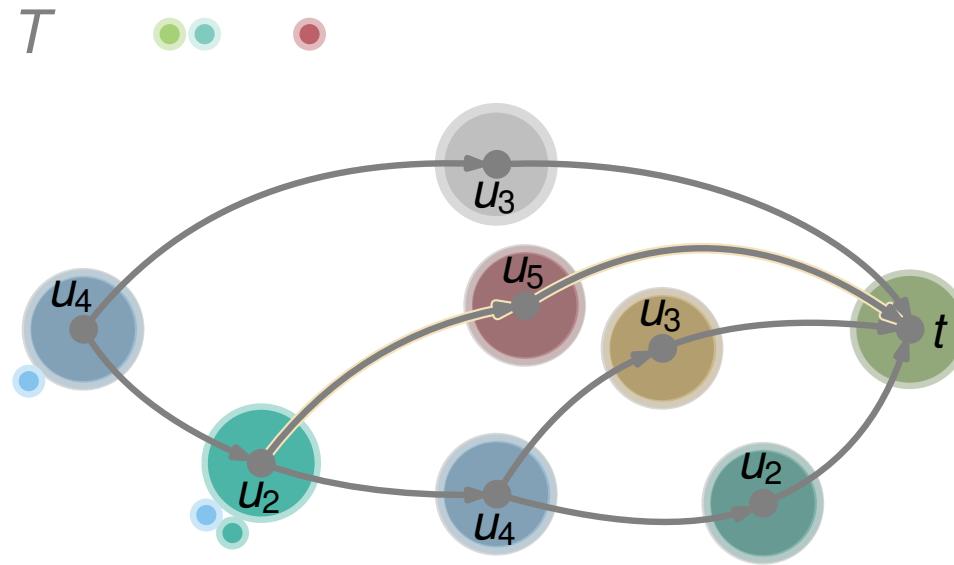


But: n colors \Rightarrow not a polynomial time algorithm(?)

Rainbow Paths – FPT Algorithm

[Theorem 11; Uchizawa et al., 2013]

- FPT algorithm w.r.t. the number of colors

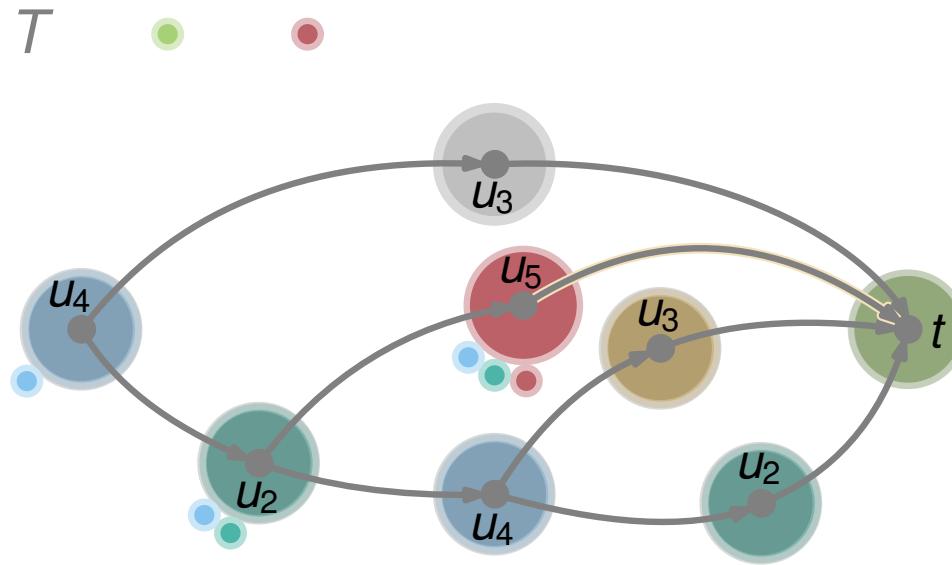


But: n colors \Rightarrow not a polynomial time algorithm(?)

Rainbow Paths – FPT Algorithm

[Theorem 11; Uchizawa et al., 2013]

- FPT algorithm w.r.t. the number of colors

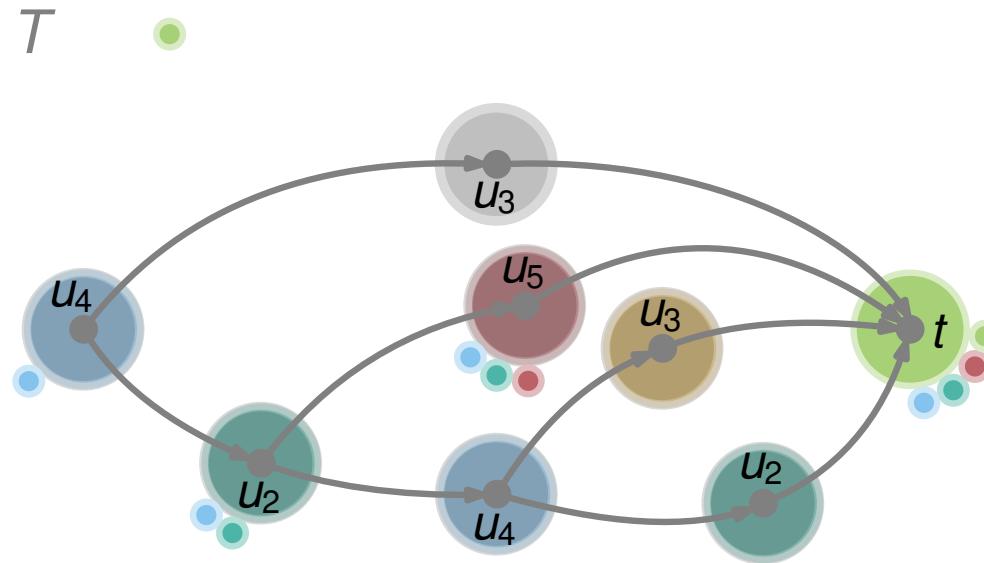


But: n colors \Rightarrow not a polynomial time algorithm(?)

Rainbow Paths – FPT Algorithm

[Theorem 11; Uchizawa et al., 2013]

- FPT algorithm w.r.t. the number of colors

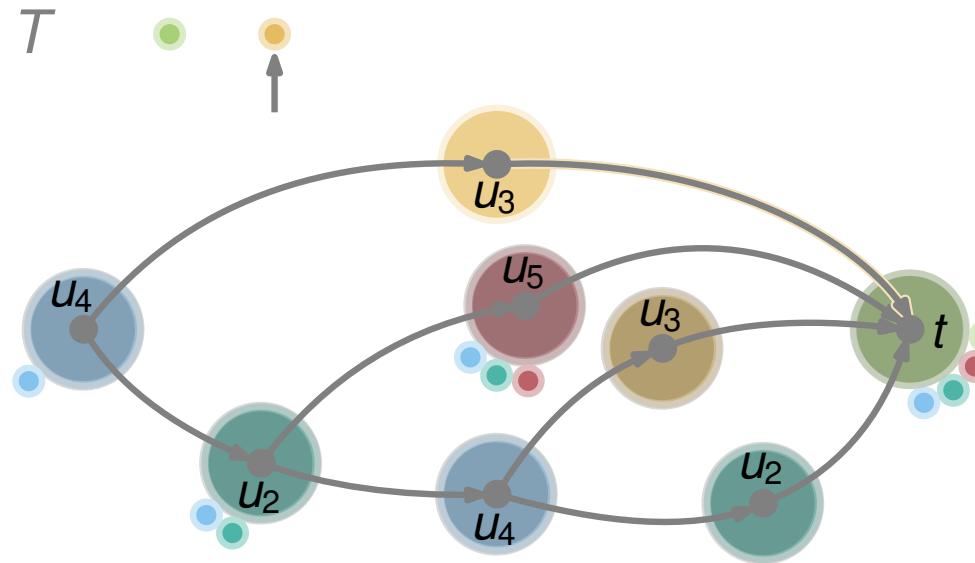


But: n colors \Rightarrow not a polynomial time algorithm(?)

Rainbow Paths – FPT Algorithm

[Theorem 11; Uchizawa et al., 2013]

- FPT algorithm w.r.t. the number of colors

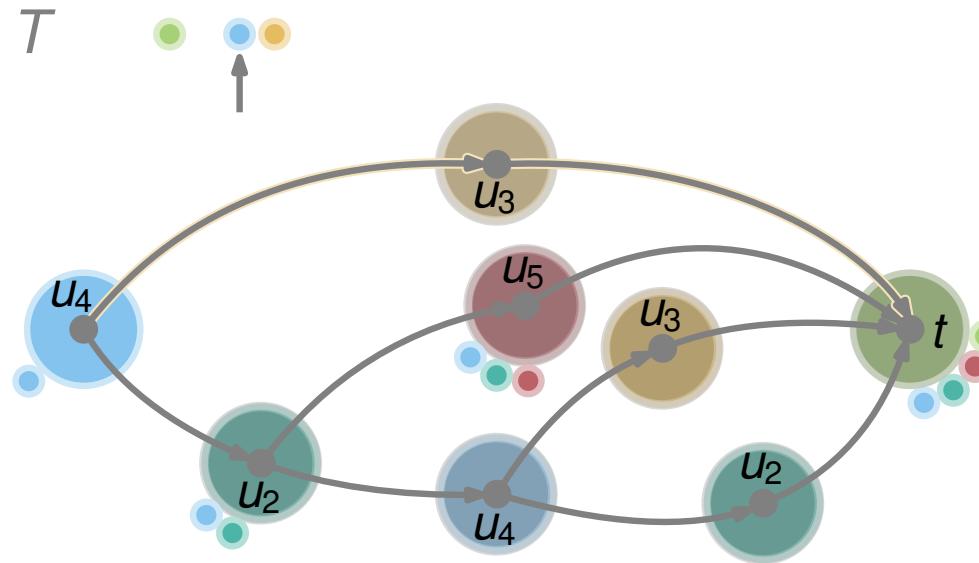


But: n colors \Rightarrow not a polynomial time algorithm(?)

Rainbow Paths – FPT Algorithm

[Theorem 11; Uchizawa et al., 2013]

- FPT algorithm w.r.t. the number of colors

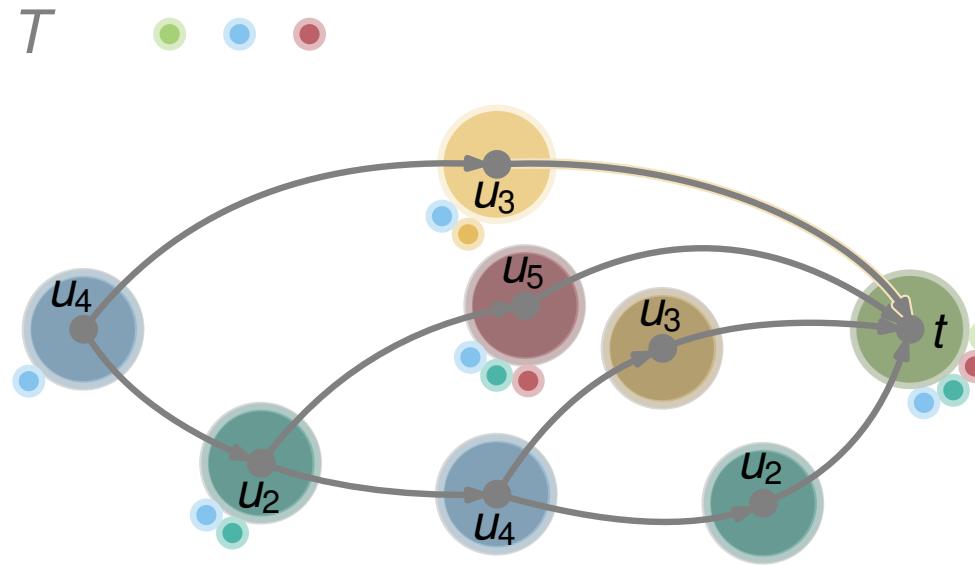


But: n colors \Rightarrow not a polynomial time algorithm(?)

Rainbow Paths – FPT Algorithm

[Theorem 11; Uchizawa et al., 2013]

- FPT algorithm w.r.t. the number of colors

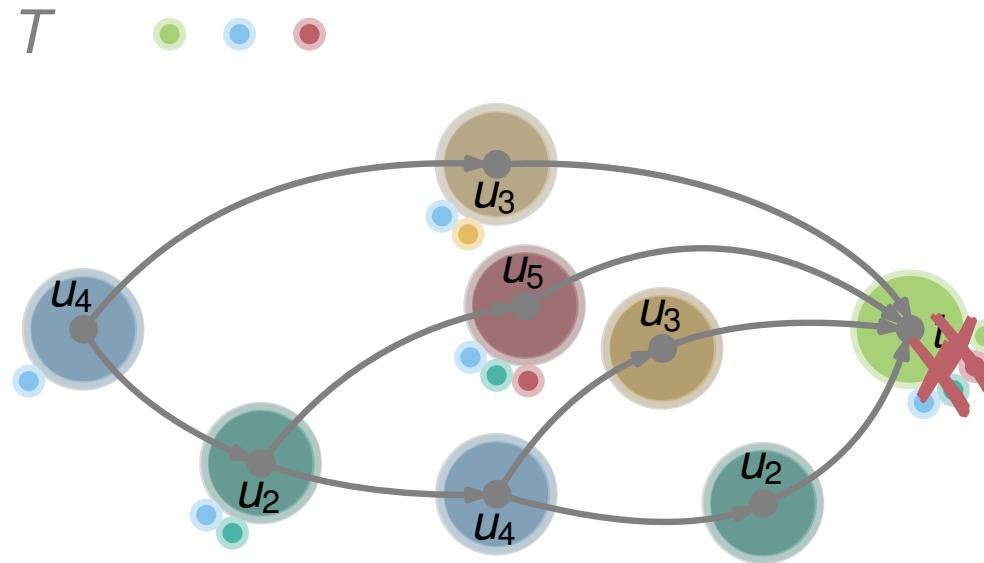


But: n colors \Rightarrow not a polynomial time algorithm(?)

Rainbow Paths – FPT Algorithm

[Theorem 11; Uchizawa et al., 2013]

- FPT algorithm w.r.t. the number of colors

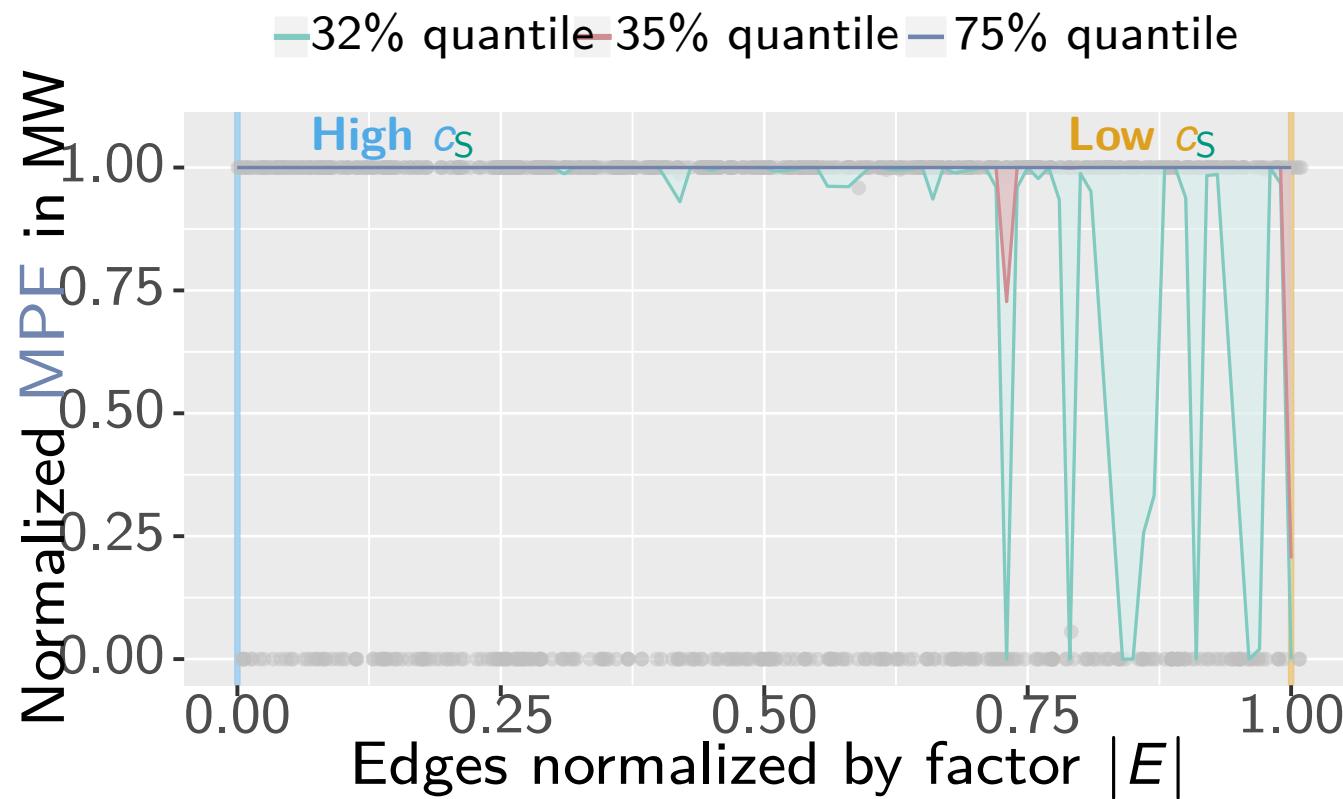


But: n colors \Rightarrow not a polynomial time algorithm(?)

Simulations

[page 349; Grastien et al., 2018]

- Simulations on NESTA benchmark sets that are more realistic than the IEEE benchmark sets, e.g., with regards to thermal line limits



The MPF decreases mainly for edges having a small centrality c_S .

Simulations

[page 349; Grastien et al., 2018]

- Simulations on NESTA benchmark sets that are more realistic than the IEEE benchmark sets, e.g., with regards to thermal line limits

— 32% quantile — 35% quantile — 75% quantile

On **general networks** the *switching centrality* $c_S: E \rightarrow \mathbb{R}_{\geq 0}$ is defined by

$$c_S(e) := \frac{1}{m_B} \sum_{s \in V} \sum_{t \in V \setminus \{s\}} \frac{\sigma_{\text{DTP}}(s, t, e)}{\sigma_{\text{DTP}}(s, t)},$$

where $\sigma_{\text{DTP}}(s, t, e)$ is the number of DTP-paths between s and t that use e , $\sigma_{\text{DTP}}(s, t)$ is the total number of DTP-paths from s to t and $m_B = |V|(|V| - 1)$.



The MPF decreases mainly for edges having a small centrality c_S .

Overview of the MTSF Results

	Graph Structure	Complexity	Algorithm
complexity ↓	penrose-minor-free graphs	polynomial-time solvable	DTP
	series-parallel graphs	NP-hard	
	cacti with max degree of 3	NP-hard [Lehmann et al., 2014]	2-approx.
	2-level trees	NP-hard [Lehmann et al., 2014]	
	planar graphs with max degree of 3	strongly NP-hard [Lehmann et al., 2014]	
	arbitrary graphs	non-APX [Lehmann et al., 2014]	

Overview of the MTSF Results

	Graph Structure	Complexity	Algorithm
complexity $ V_G = 2, V_D = 2$	penrose-minor-free graphs	polynomial-time solvable	DTP
	series-parallel graphs	NP-hard	
	cacti with max degree of 3	NP-hard [Lehmann et al., 2014]	2-approx.
	2-level trees	NP-hard [Lehmann et al., 2014]	
	planar graphs with max degree of 3	strongly NP-hard [Lehmann et al., 2014]	
	arbitrary graphs	non-APX [Lehmann et al., 2014]	

Subset Sum Problem (SSP)

Decision Problem Subset Sum (SSP)

Instance: A finite set of numbers $W = \{w_1, w_2, \dots, w_n\}$ with $w_i \in \mathbb{N}$ and $k \in \mathbb{N}$.

Question: Is there a set of elements $x_1, x_2, \dots, x_n \in \{0, 1\}$ such that $\sum_{j=1}^n w_j x_j = k$?

Subset Sum Problem (SSP)

Decision Problem Subset Sum (SSP)

Instance: A finite set of numbers $W = \{w_1, w_2, \dots, w_n\}$ with $w_i \in \mathbb{N}$ and $k \in \mathbb{N}$.

Question: Is there a set of elements $x_1, x_2, \dots, x_n \in \{0, 1\}$ such that $\sum_{j=1}^n w_j x_j = k$?

Example Instance

- $W = \{1, 2, 3, 7, 37, 99\}$
- $k = 42$

Subset Sum Problem (SSP)

Decision Problem Subset Sum (SSP)

Instance: A finite set of numbers $W = \{w_1, w_2, \dots, w_n\}$ with $w_i \in \mathbb{N}$ and $k \in \mathbb{N}$.

Question: Is there a set of elements $x_1, x_2, \dots, x_n \in \{0, 1\}$ such that $\sum_{j=1}^n w_j x_j = k$?

Example Instance

- $W = \{1, 2, 3, 7, 37, 99\}$
- $k = 42$

Solution

- $X = \{2, 3, 37\}$

OTS is NP-complete on Series-parallel Graphs Single-Source-Sink

[Kocuk et al., 2016]

Theorem 4 [page 926; Kocuk et al., 2016]

The feasibility version of OTS is NP-complete even if $\mathcal{N} = (G, V_G, V_D, \text{cap}, b, d)$ is a series-parallel graph with one generator $|V_G| = 1$, and one consumer $|V_D| = 1$.

OTS is NP-complete on Series-parallel Graphs Single-Source-Sink

[Kocuk et al., 2016]

Theorem 4 [page 926; Kocuk et al., 2016]

The feasibility version of OTS is NP-complete even if $\mathcal{N} = (G, V_G, V_D, \text{cap}, b, d)$ is a series-parallel graph with one generator  , and one consumer  .



OTS is NP-complete on Series-parallel Graphs

Single-Source-Sink

[Kocuk et al., 2016]

Instance:

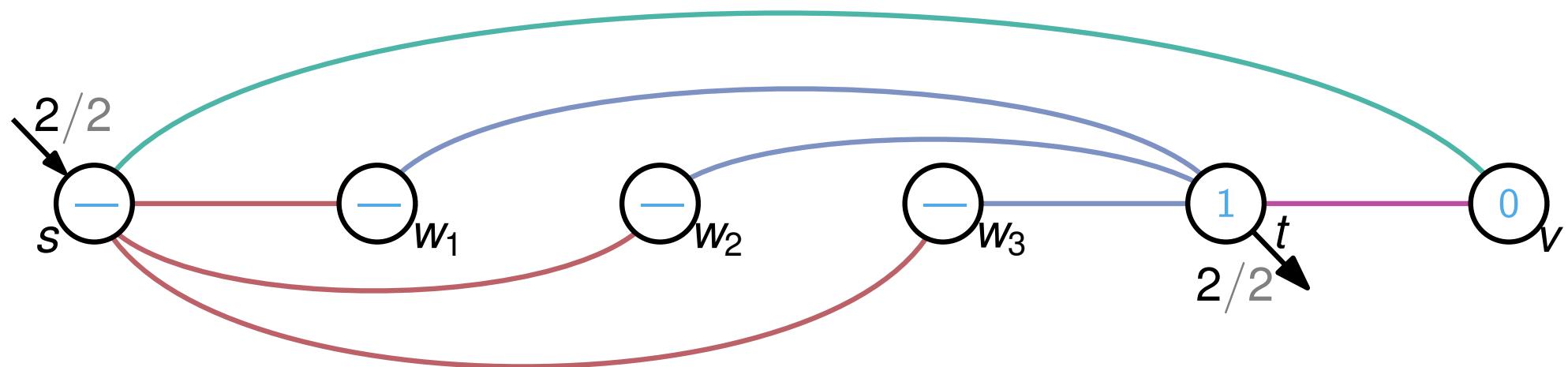
vertices	<ul style="list-style-type: none">■ $n + 3$ vertices $\{0, 1, \dots, n, n + 1, n + 2\}$
edges	<ul style="list-style-type: none">■ $(0, i),$ $(i, n + 1),$ $(n + 1, n + 2),$ $(0, n + 2).$ $\forall i \in \{1, \dots, n\}$
capacities	<ul style="list-style-type: none">■ $\frac{w_i}{k}$ for $(0, i), (i, n + 1),$ 1 for $(n + 1, n + 2), (0, n + 2),$ $\forall i \in \{1, \dots, n\}$
susceptances	<ul style="list-style-type: none">■ $2w_i$ for $(0, i), (i, n + 1),$ 1 for $(n + 1, n + 2),$ $\frac{k}{k+1}$ for $(0, n + 2)$ $\forall i \in \{1, \dots, n\}$
generation / load	<ul style="list-style-type: none">■ generation of 2 at bus 0, consumption of 2 at bus $n + 1$

OTS is NP-complete on Series-parallel Graphs

Single-Source-Sink

[Kocuk et al., 2016]

- Instance: $n = 3$, $W = \{1, 2, 3\}$, and $k = 3$.

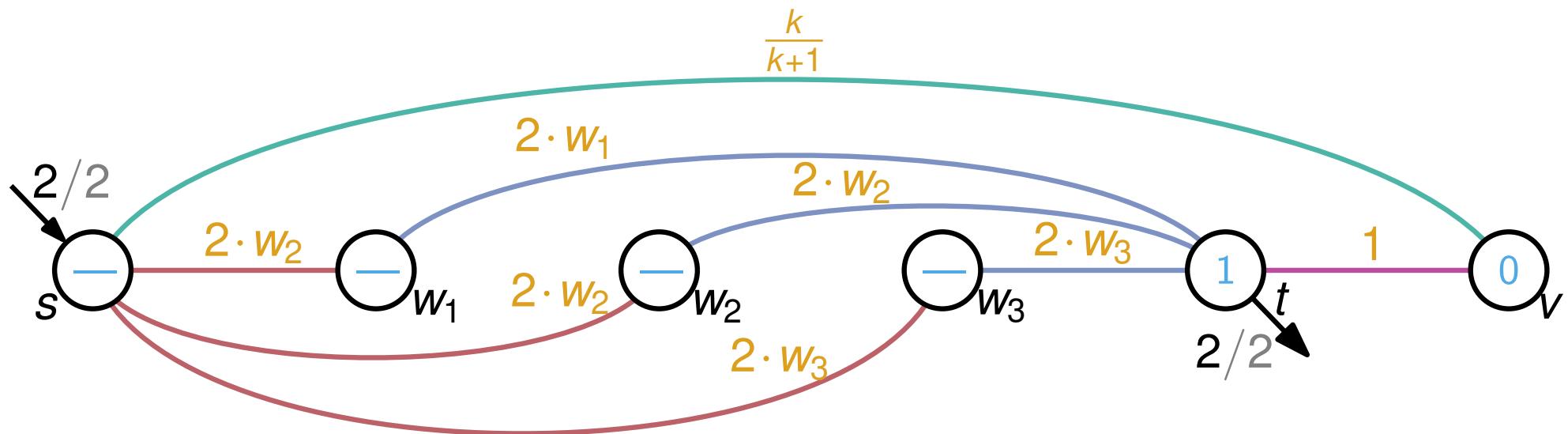


OTS is NP-complete on Series-parallel Graphs

Single-Source-Sink

[Kocuk et al., 2016]

- Instance: $n = 3$, $W = \{1, 2, 3\}$, and $k = 3$.

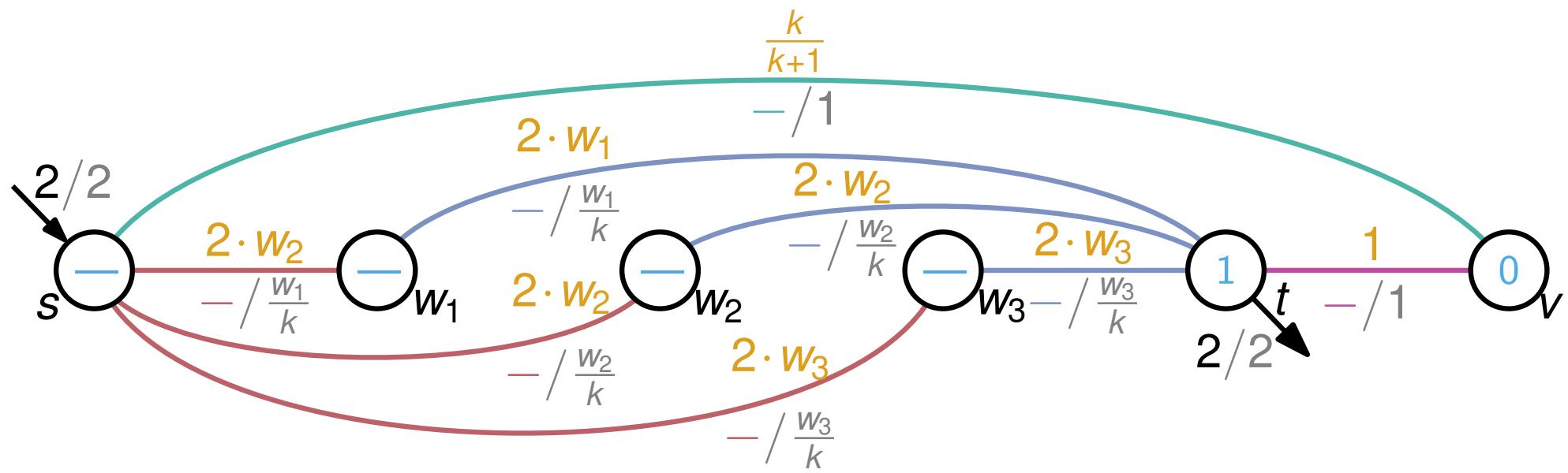


OTS is NP-complete on Series-parallel Graphs

Single-Source-Sink

[Kocuk et al., 2016]

- Instance: $n = 3$, $W = \{1, 2, 3\}$, and $k = 3$.

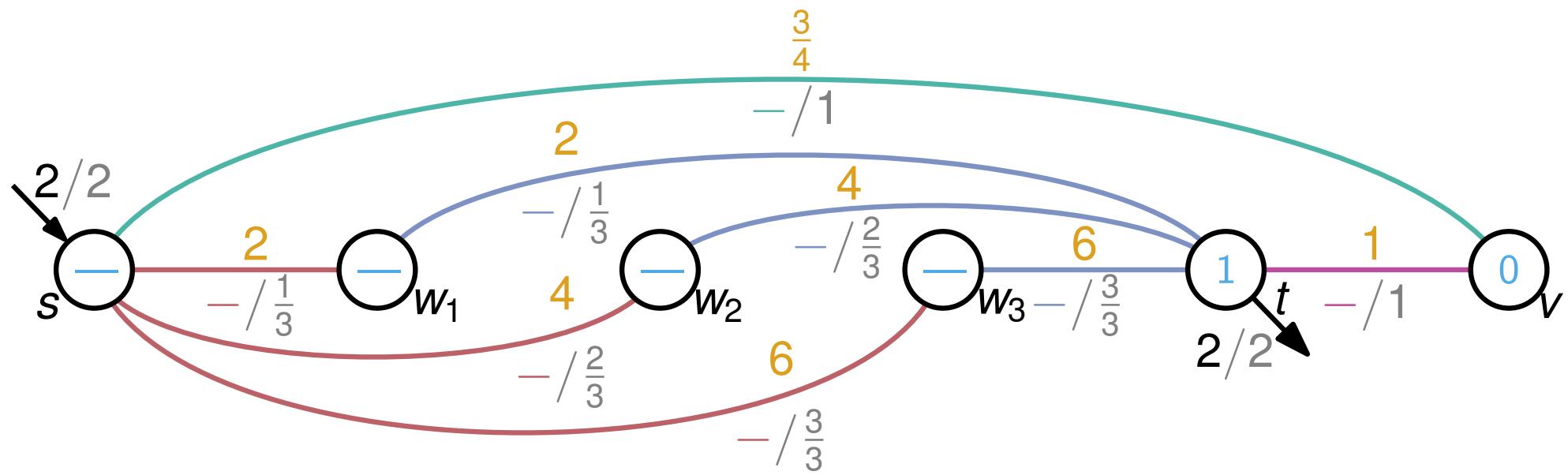


OTS is NP-complete on Series-parallel Graphs

Single-Source-Sink

[Kocuk et al., 2016]

- Instance: $n = 3$, $W = \{1, 2, 3\}$, and $k = 3$.

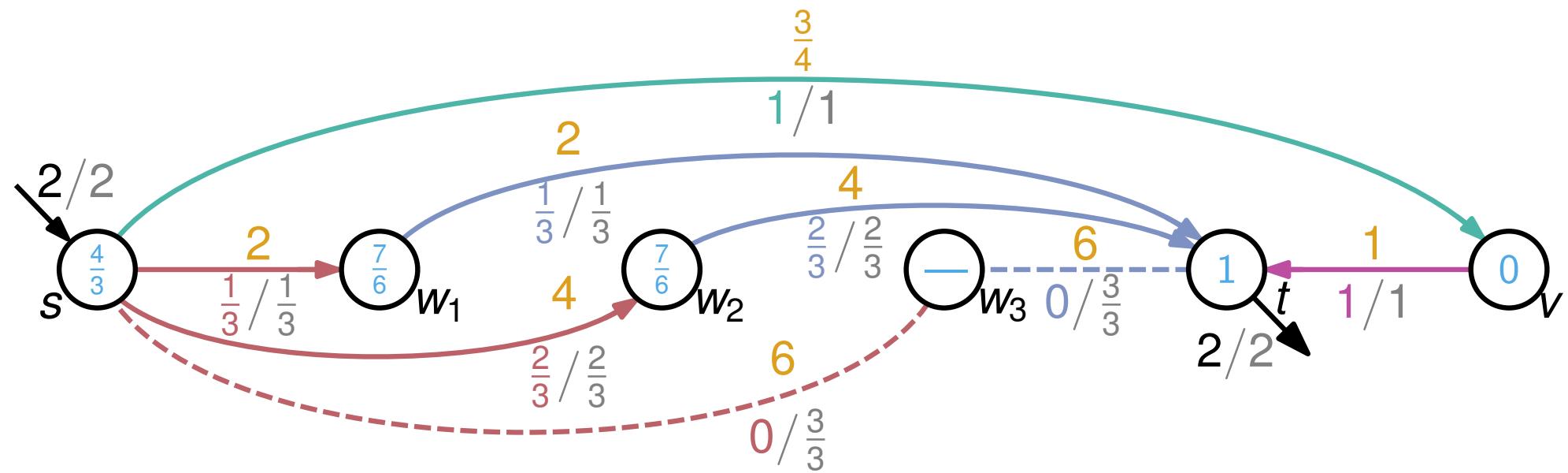


OTS is NP-complete on Series-parallel Graphs

Single-Source-Sink

[Kocuk et al., 2016]

- Instance: $n = 3$, $W = \{1, 2, 3\}$, and $k = 3$.

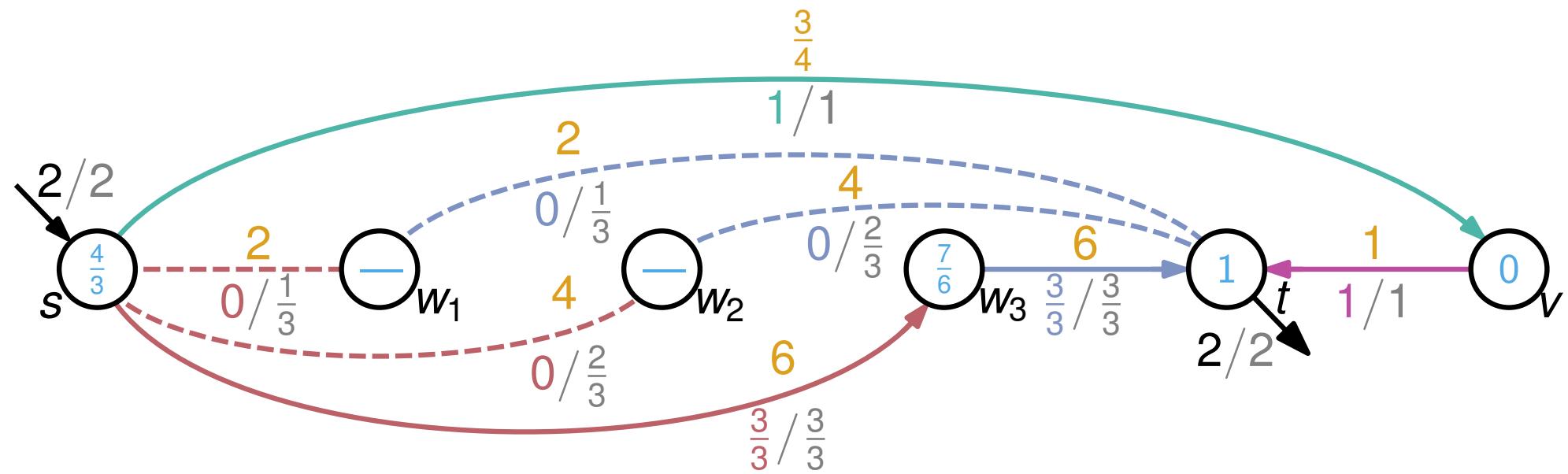


OTS is NP-complete on Series-parallel Graphs

Single-Source-Sink

[Kocuk et al., 2016]

- Instance: $n = 3$, $W = \{1, 2, 3\}$, and $k = 3$.



The feasibility version of OTS is NP-complete on an $s-t$ series-parallel graph.

OTS is NP-complete on Series-parallel Graphs

[Kocuk et al., 2016]

- Size of the OTS-instance is polynomial in the size of the SSP-instance
- Graph G is series-parallel
- Only one generator and one demand

k -MTSF is NP-complete on Series-parallel Graphs Single-Source-Sink and Unit-Capacities

[Grastien et al., 2018]

Theorem 5 [page 355; Grastien et al., 2018]

The k -MTSF problem is NP-complete even if there is only one generator $|V_G| = 1$ and one consumer $|V_D| = 1$ in the network $\mathcal{N} = (G, V_G, V_D, \text{cap}, b, d)$ where the underlying graph G is series-parallel and all edge capacities are $\text{cap}(e) = 1$ for all e in E .

k-MTSF is NP-complete on Series-parallel Graphs Single-Source-Sink and Unit-Capacities [Grastien et al., 2018]

Theorem 5 [page 355; Grastien et al., 2018]

The ***k-MTSF*** problem is NP-complete even if there is only one generator  and one consumer  in the network $\mathcal{N} = (G, V_G, V_D, \text{cap}, b, d)$ where the underlying graph G is  and all edge capacities are $\text{cap}(e) = 1$ for all e in E .

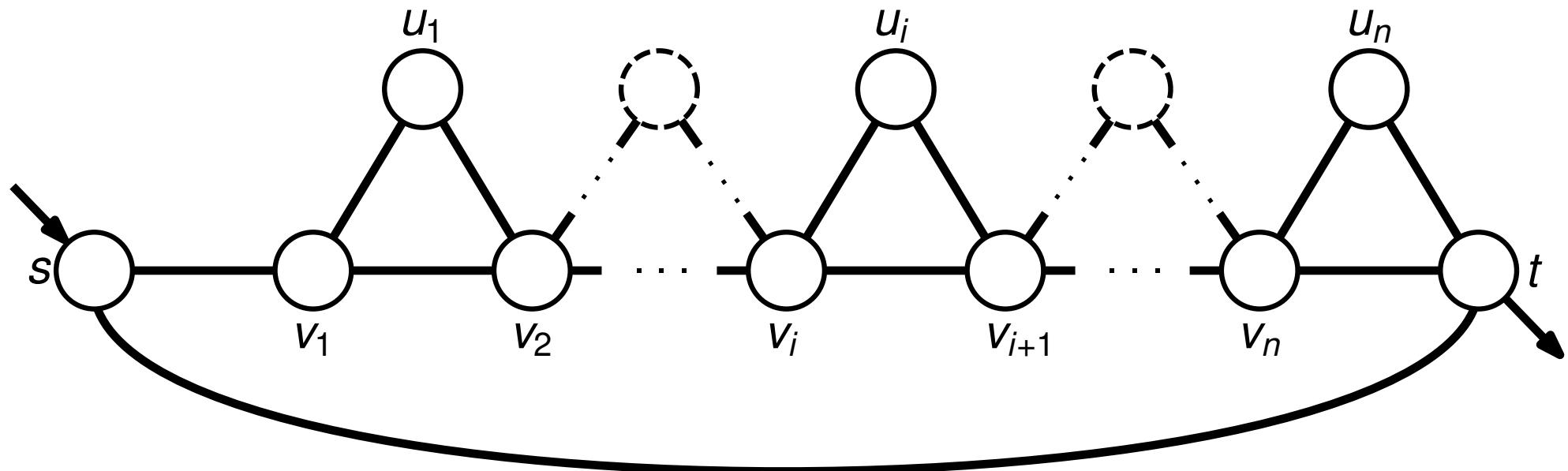
***k*-MTSF is NP-complete on Series-parallel Graphs Single-Source-Sink and Unit-Capacities**

[Grastien et al., 2018]

$$\text{cap}(i, j) := 1 \quad \forall (i, j) \in E$$

$$|V_G| := |V_D| := 1$$

$$W = \{w_1, w_2, \dots, w_n\}, n := |W|$$



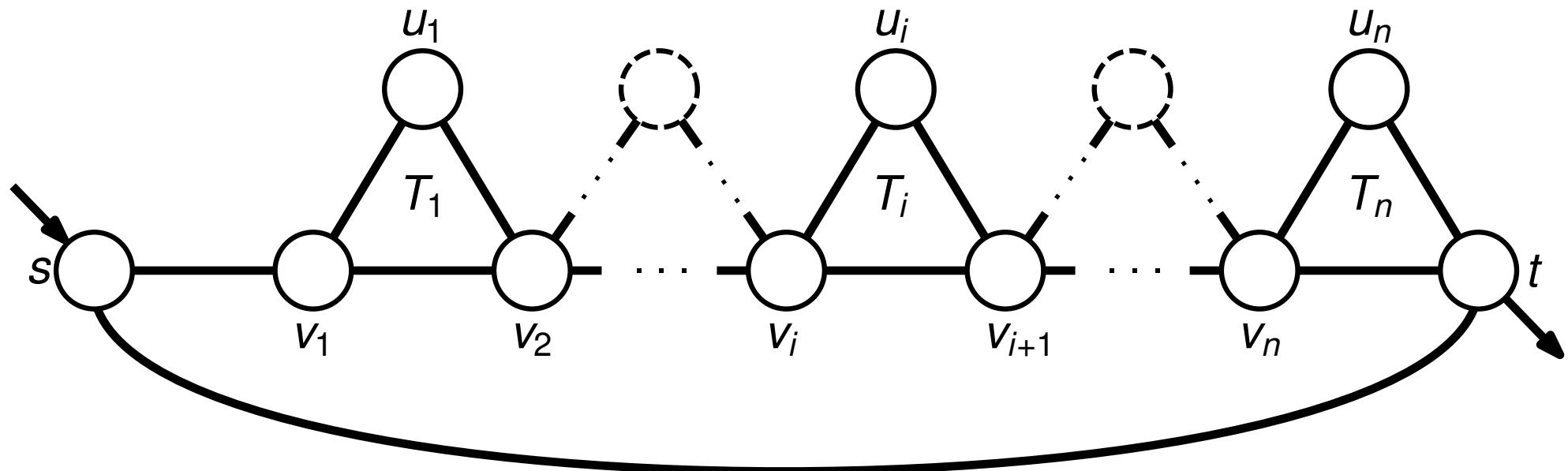
***k*-MTSF is NP-complete on Series-parallel Graphs Single-Source-Sink and Unit-Capacities**

[Grastien et al., 2018]

$$\text{cap}(i, j) := 1 \quad \forall (i, j) \in E$$

$$|V_G| := |V_D| := 1$$

$$W = \{w_1, w_2, \dots, w_n\}, n := |W|$$



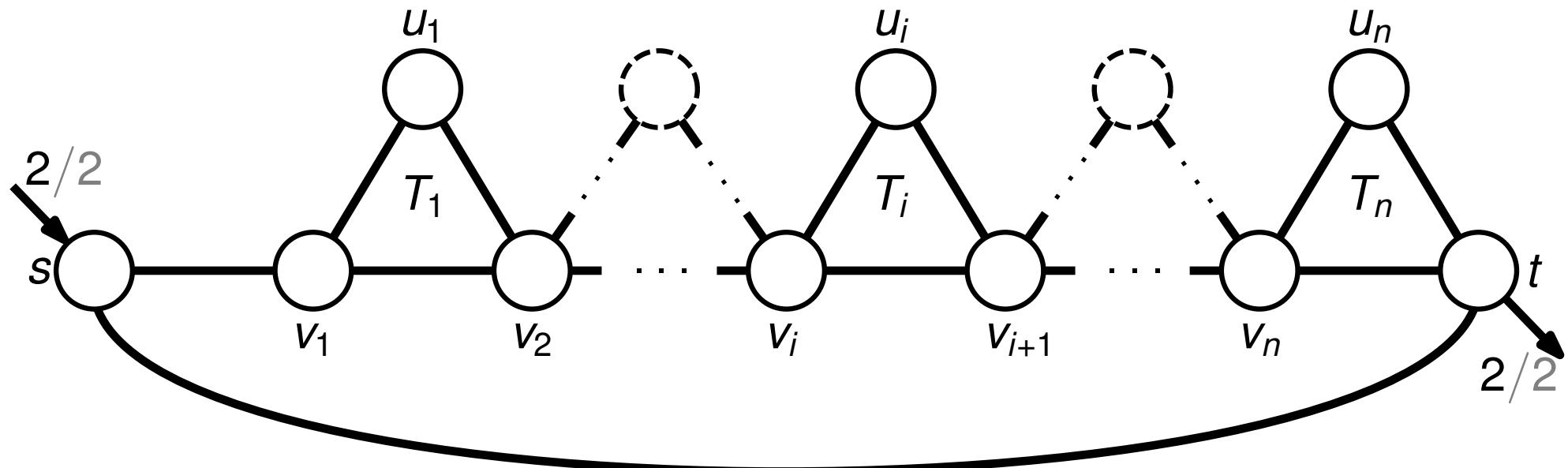
***k*-MTSF is NP-complete on Series-parallel Graphs Single-Source-Sink and Unit-Capacities**

[Grastien et al., 2018]

$$\text{cap}(i, j) := 1 \quad \forall (i, j) \in E$$

$$|V_G| := |V_D| := 1$$

$$W = \{w_1, w_2, \dots, w_n\}, n := |W|$$



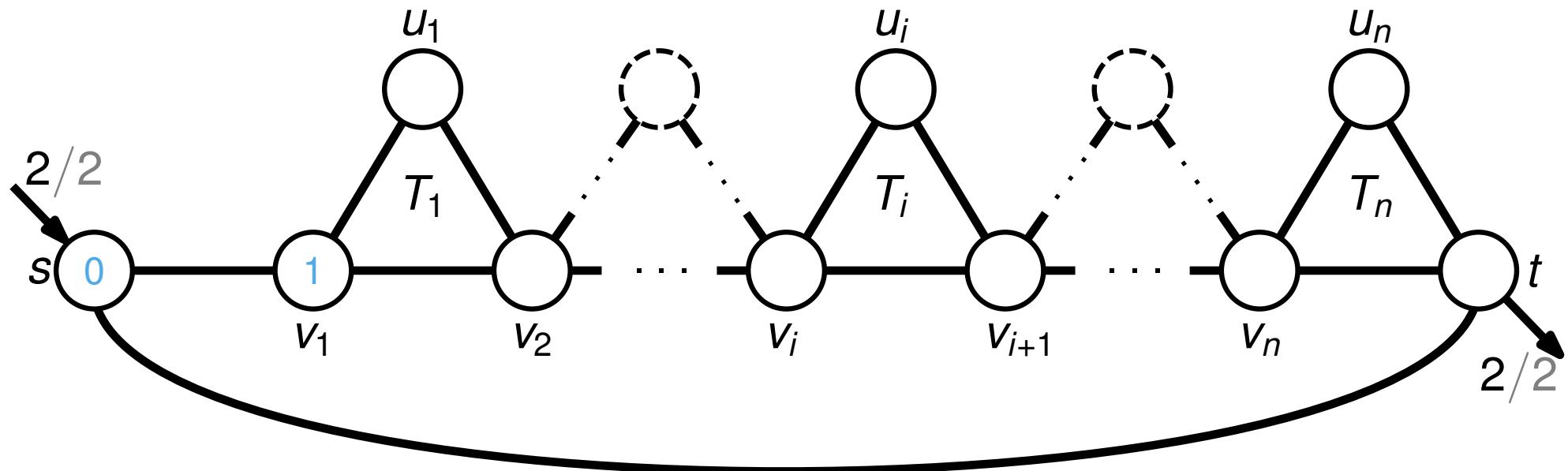
***k*-MTSF is NP-complete on Series-parallel Graphs Single-Source-Sink and Unit-Capacities**

[Grastien et al., 2018]

$$\text{cap}(i, j) := 1 \quad \forall (i, j) \in E$$

$$|V_G| := |V_D| := 1$$

$$W = \{w_1, w_2, \dots, w_n\}, n := |W|$$



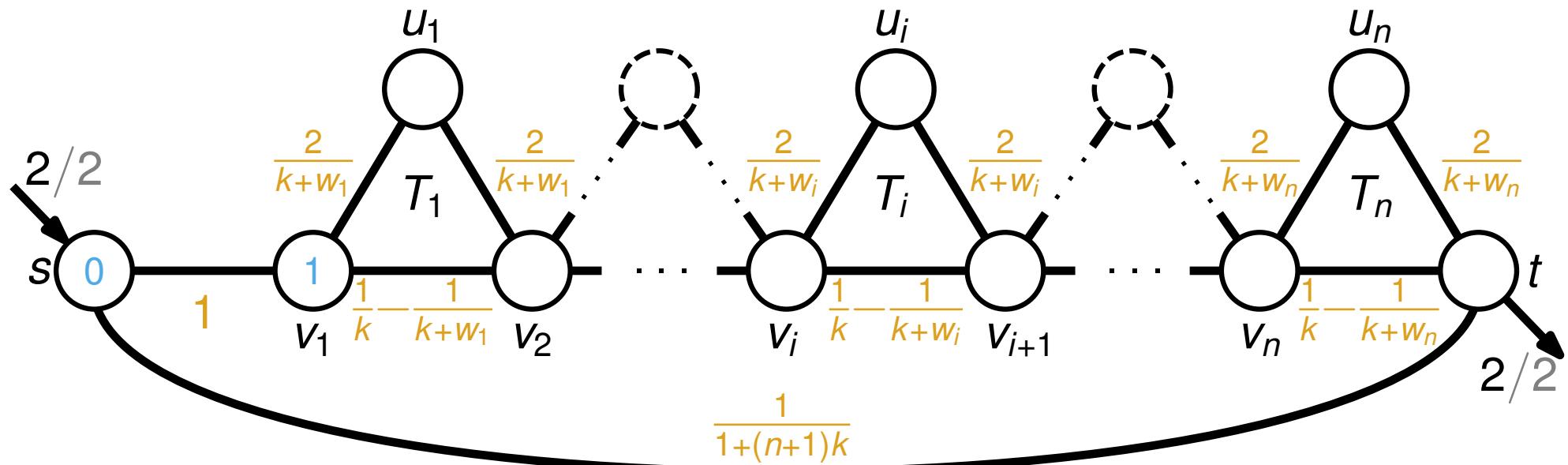
k -MTSF is NP-complete on Series-parallel Graphs Single-Source-Sink and Unit-Capacities

[Grastien et al., 2018]

$$\text{cap}(i, j) := 1 \quad \forall (i, j) \in E$$

$$|V_G| := |V_D| := 1$$

$$W = \{w_1, w_2, \dots, w_n\}, n := |W|$$



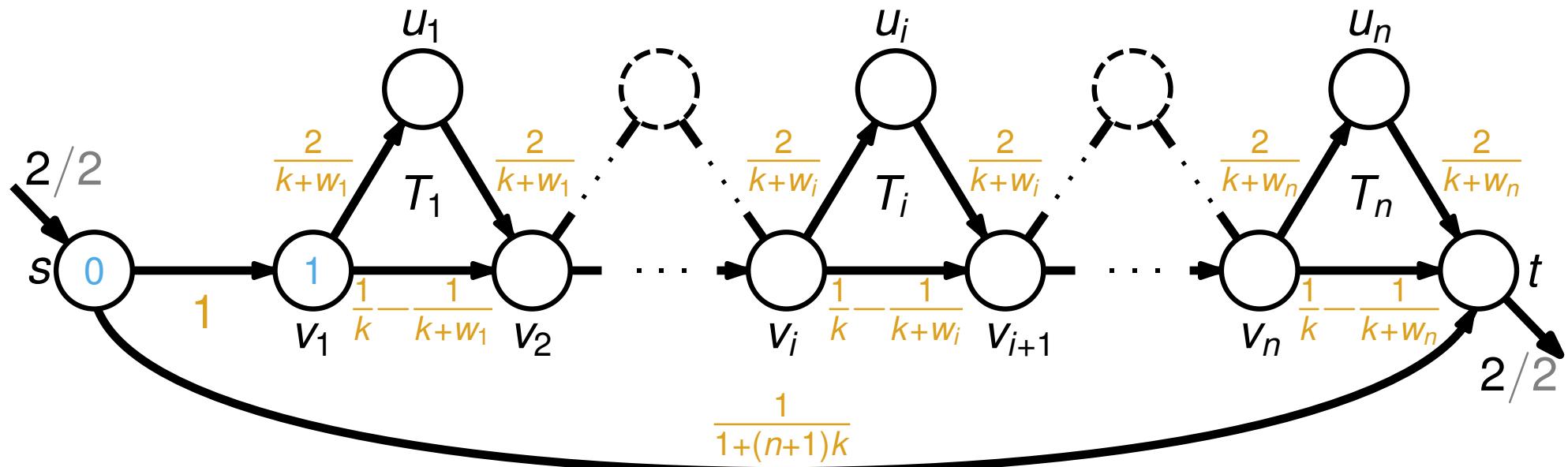
k -MTSF is NP-complete on Series-parallel Graphs Single-Source-Sink and Unit-Capacities

[Grastien et al., 2018]

$$\text{cap}(i, j) := 1 \quad \forall (i, j) \in E$$

$$|V_G| := |V_D| := 1$$

$$W = \{w_1, w_2, \dots, w_n\}, n := |W|$$



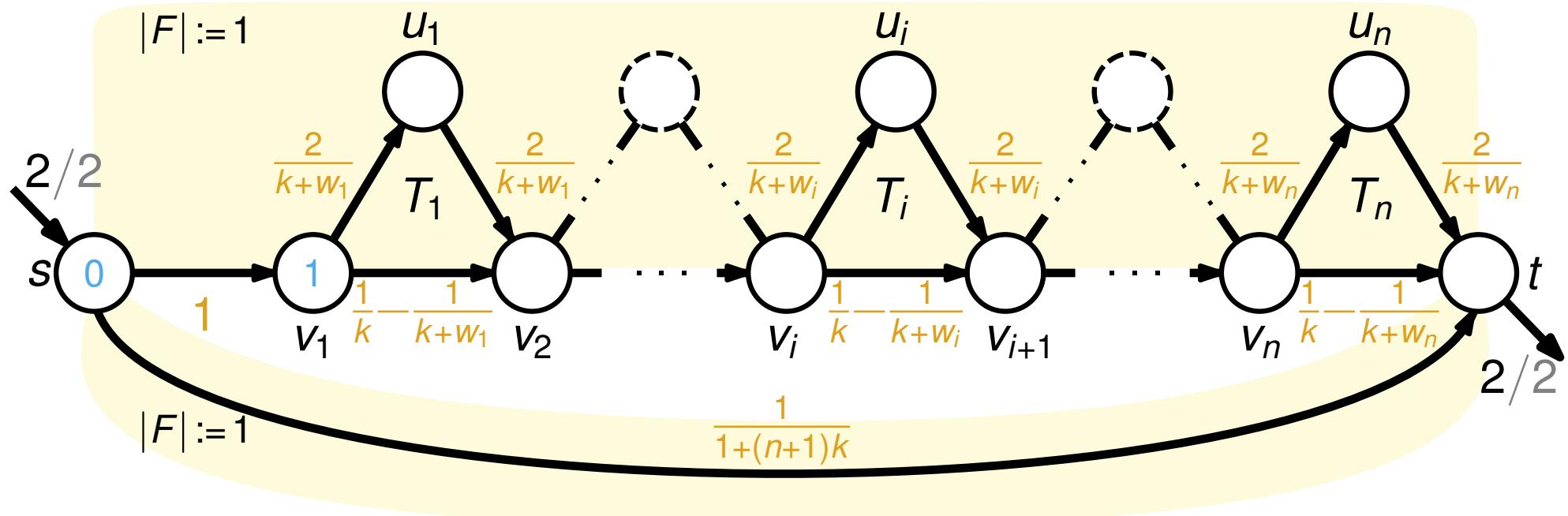
k -MTSF is NP-complete on Series-parallel Graphs Single-Source-Sink and Unit-Capacities

[Grastien et al., 2018]

$$\text{cap}(i, j) := 1 \quad \forall (i, j) \in E$$

$$|V_G| := |V_D| := 1$$

$$W = \{w_1, w_2, \dots, w_n\}, n := |W|$$



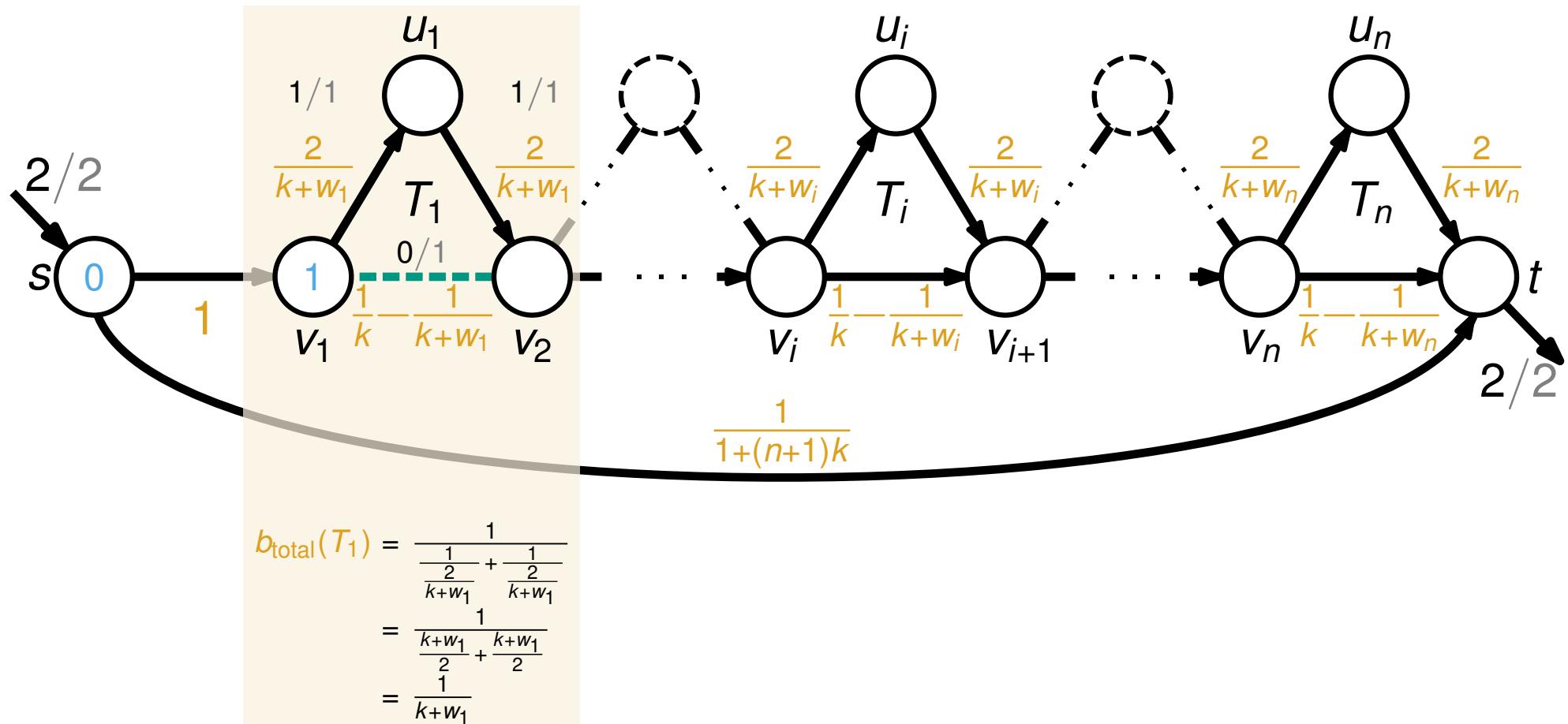
k -MTSF is NP-complete on Series-parallel Graphs Single-Source-Sink and Unit-Capacities

[Grastien et al., 2018]

$$\text{cap}(i, j) := 1 \quad \forall (i, j) \in E$$

$$|V_G| := |V_D| := 1$$

$$W = \{w_1, w_2, \dots, w_n\}, n := |W|$$



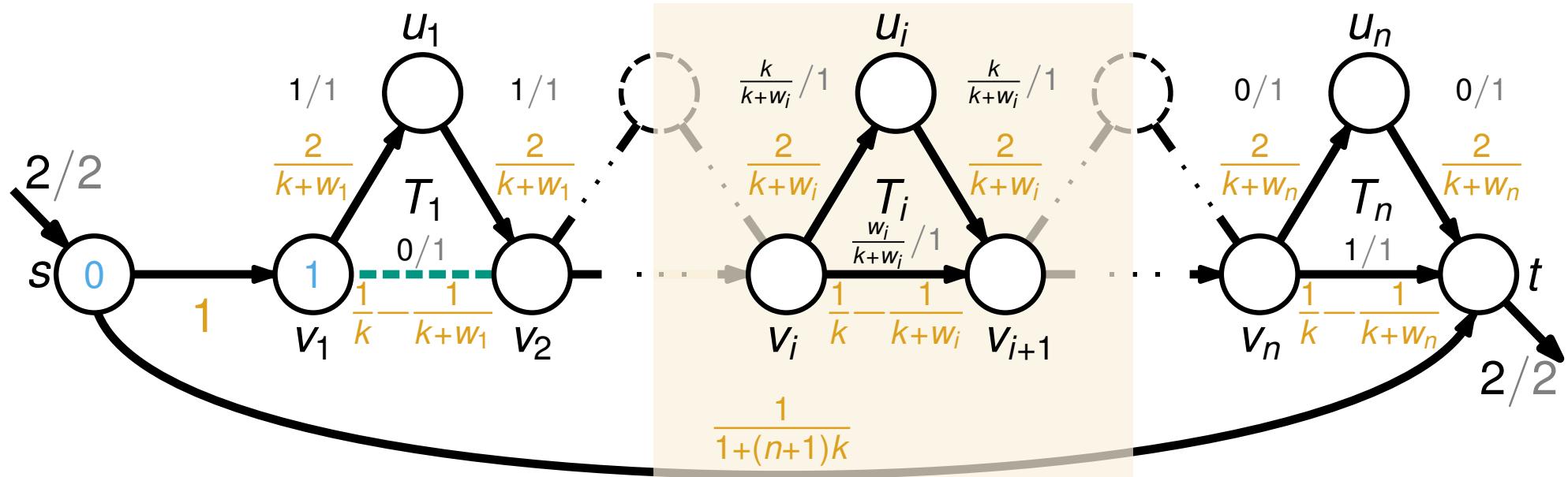
k -MTSF is NP-complete on Series-parallel Graphs Single-Source-Sink and Unit-Capacities

[Grastien et al., 2018]

$$\text{cap}(i, j) := 1 \quad \forall (i, j) \in E$$

$$|V_G| := |V_D| := 1$$

$$W = \{w_1, w_2, \dots, w_n\}, n := |W|$$



$$\begin{aligned} b_{\text{total}}(T_1) &= \frac{1}{\frac{1}{\frac{2}{k+w_1}} + \frac{1}{\frac{2}{k+w_1}}} \\ &= \frac{1}{\frac{k+w_1}{2} + \frac{k+w_1}{2}} \\ &= \frac{1}{k+w_1} \end{aligned}$$

$$\begin{aligned} b_{\text{total}}(T_i) &= \frac{1}{\frac{1}{\frac{2}{k+w_i}} + \frac{1}{\frac{2}{k+w_i}}} + \frac{1}{k} - \frac{1}{k+w_i} \\ &= \frac{1}{k+w_i} + \frac{1}{k} - \frac{1}{k+w_i} \\ &= \frac{1}{k} \end{aligned}$$

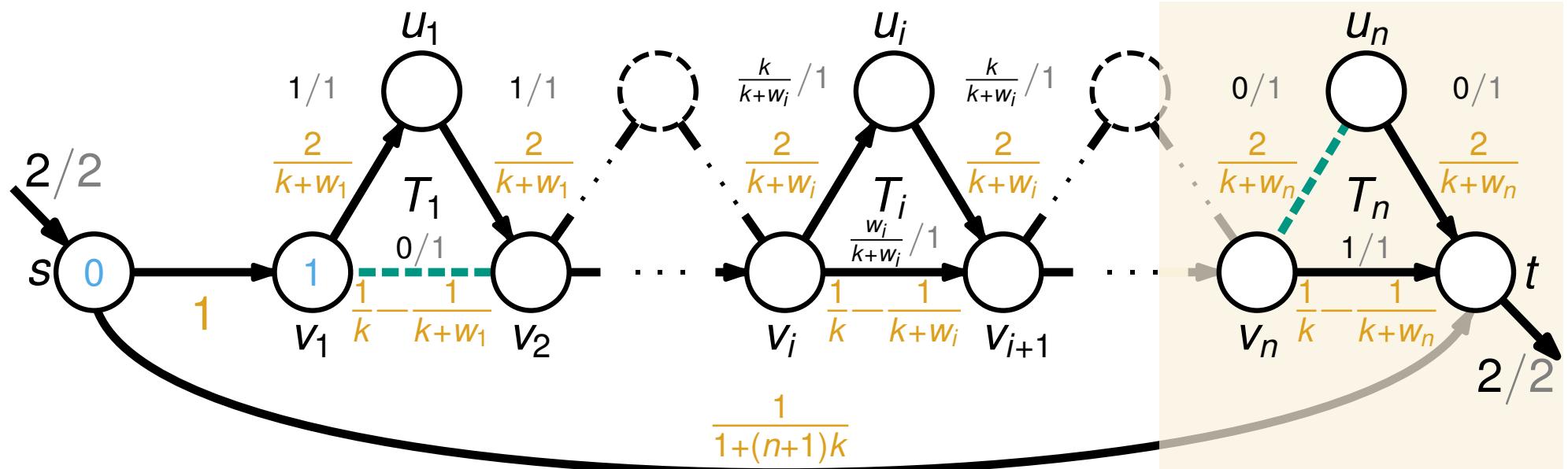
k -MTSF is NP-complete on Series-parallel Graphs Single-Source-Sink and Unit-Capacities

[Grastien et al., 2018]

$$\text{cap}(i, j) := 1 \quad \forall (i, j) \in E$$

$$|V_G| := |V_D| := 1$$

$$W = \{w_1, w_2, \dots, w_n\}, n := |W|$$



$$\begin{aligned} b_{\text{total}}(T_1) &= \frac{1}{\frac{1}{\frac{2}{k+w_1}} + \frac{1}{\frac{2}{k+w_1}}} \\ &= \frac{1}{\frac{k+w_1}{2} + \frac{k+w_1}{2}} \\ &= \frac{1}{k+w_1} \end{aligned}$$

$$\begin{aligned} b_{\text{total}}(T_i) &= \frac{1}{\frac{1}{\frac{2}{k+w_i}} + \frac{1}{\frac{2}{k+w_i}}} + \frac{1}{k} - \frac{1}{k+w_i} \\ &= \frac{1}{k+w_i} + \frac{1}{k} - \frac{1}{k+w_i} \\ &= \frac{1}{k} \end{aligned}$$

$$\begin{aligned} b_{\text{total}}(T_n) &= \frac{1}{k} - \frac{1}{k+w_n} \\ &= \frac{k+w_n}{k(k+w_n)} - \frac{k}{k(k+w_n)} \\ &= \frac{w_n}{k(k+w_n)} \end{aligned}$$

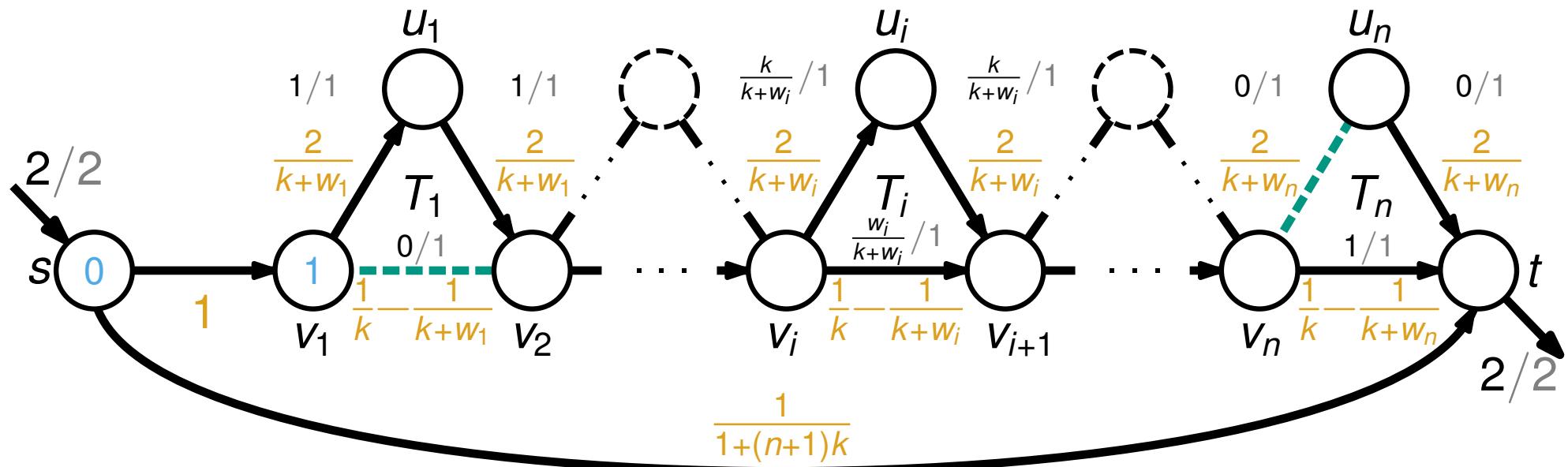
k -MTSF is NP-complete on Series-parallel Graphs Single-Source-Sink and Unit-Capacities

[Grastien et al., 2018]

$$\text{cap}(i, j) := 1 \quad \forall (i, j) \in E$$

$$|V_G| := |V_D| := 1$$

$$W = \{w_1, w_2, \dots, w_n\}, n := |W|$$



$$b_{\text{total}}(T_1) = \frac{1}{k+w_1}$$

$$b_{\text{total}}(T_i) = \frac{1}{k}$$

$$b_{\text{total}}(T_n) = \frac{w_n}{k(k+w_n)}$$

$$\Delta\theta^V(v_1, v_2) = k + w_1$$

$$\Delta\theta^V(v_i, v_{i+1}) = k$$

$$\begin{aligned} \Delta\theta^V(v_n, t) &= \frac{k(k+w_n)}{w_n} = \frac{k^2}{w_n} + k \\ &> 2k \end{aligned}$$

k -MTSF is NP-complete on Series-parallel Graphs Single-Source-Sink and Unit-Capacities

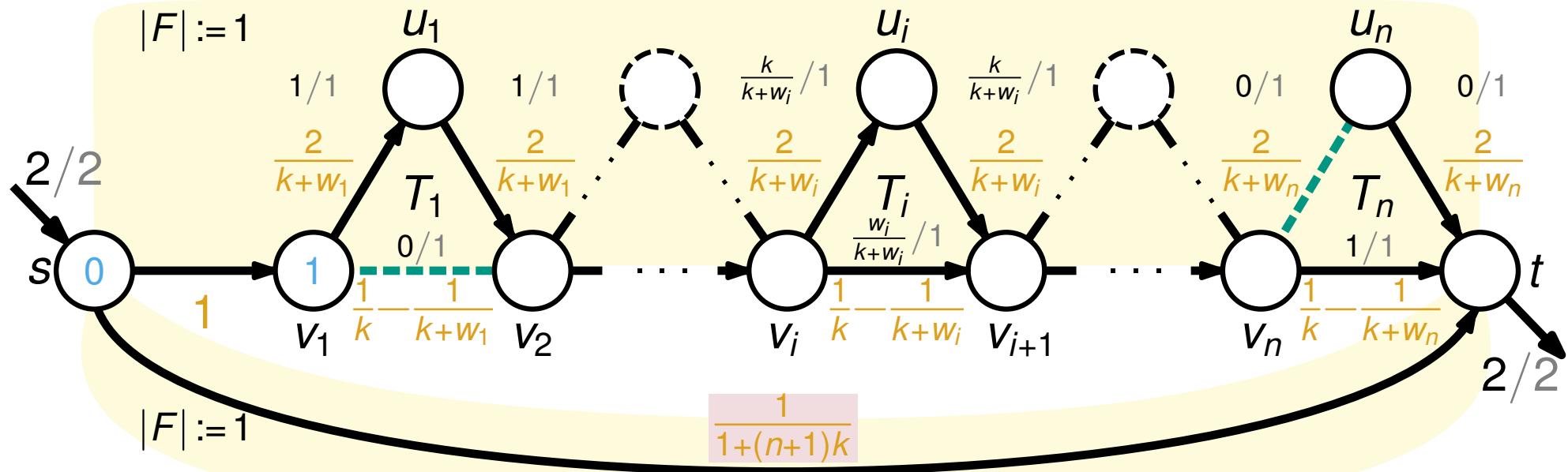
[Grastien et al., 2018]

$$\text{cap}(i, j) := 1 \quad \forall (i, j) \in E$$

$$|V_G| := |V_D| := 1$$

$$W = \{w_1, w_2, \dots, w_n\}, n := |W|$$

$$\Delta\theta_{\text{total}}^v(s, t) > \underbrace{1 + (n - 1)k}_{\text{one } (v_i, u_i) \text{ switched}} + \underbrace{2k}_{\text{not switched}} \\ = 1 + (n + 1)k$$



$$b_{\text{total}}(T_1) = \frac{1}{k+w_1}$$

$$b_{\text{total}}(T_i) = \frac{1}{k}$$

$$b_{\text{total}}(T_n) = \frac{w_n}{k(k+w_n)}$$

$$\Delta\theta^v(v_1, v_2) = k + w_1$$

$$\Delta\theta^v(v_i, v_{i+1}) = k$$

$$\Delta\theta^v(v_n, t) = \frac{k(k+w_n)}{w_n} = \frac{k^2}{w_n} + k \\ > 2k$$

k -MTSF is NP-complete on Series-parallel Graphs Single-Source-Sink and Unit-Capacities

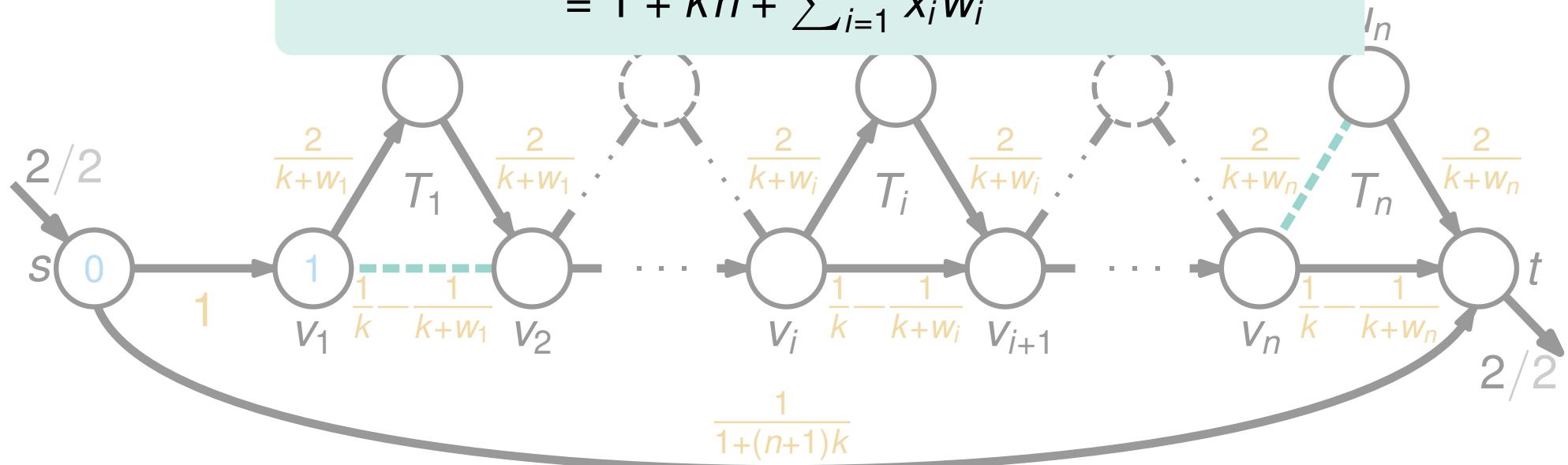
[Grastien et al., 2018]

$$\text{cap}(i, j) := 1 \quad \forall i, j$$

$$|V_G| := |V_D| := 1$$

$$W = \{w_1, w_2, \dots\}$$

$$\begin{aligned}
 1 + (n+1)k &= \Delta\theta^v(s, t) \\
 &= \Delta\theta^v(s, v_1) + \sum_{i=1}^n \Delta\theta^v(v_i, v_{i+1}) \\
 &= 1 + \sum_{i=1}^n (k + x_i w_i) \\
 &= 1 + kn + \sum_{i=1}^n x_i w_i
 \end{aligned}$$



$$b_{\text{total}}(T_1) = \frac{1}{k+w_1}$$

$$b_{\text{total}}(T_i) = \frac{1}{k}$$

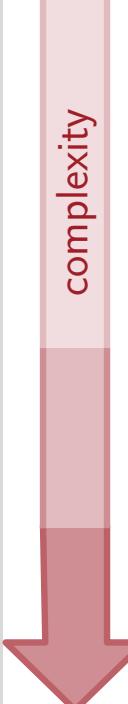
$$b_{\text{total}}(T_n) = \frac{w_n}{k(k+w_n)}$$

$$\Delta\theta^v(v_1, v_2) = k + w_1$$

$$\Delta\theta^v(v_i, v_{i+1}) = k$$

$$\begin{aligned}
 \Delta\theta^v(v_n, t) &= \frac{k(k+w_n)}{w_n} = \frac{k^2}{w_n} + k \\
 &> 2k
 \end{aligned}$$

Overview of the MTSF Results

	Graph Structure	Complexity	Algorithm
complexity 	penrose-minor-free graphs	polynomial-time solvable	DTP 
	series-parallel graphs	NP-hard	
	cacti with max degree of 3	NP-hard [Lehmann et al., 2014]	2-approx. 
	2-level trees	NP-hard [Lehmann et al., 2014]	
	planar graphs with max degree of 3	strongly NP-hard [Lehmann et al., 2014]	
	arbitrary graphs	non-APX [Lehmann et al., 2014]	

Overview of the MTSF Results

	Graph Structure	Complexity	Algorithm
one generator, one load	<p>penrose-minor-free graphs</p>  <p>series-parallel graphs</p> 	<p>polynomial-time solvable</p> <p>NP-hard</p>	 
complexity	cacti with max degree of 3	<p>NP-hard</p> <p>[Lehmann et al. 2014]</p>	
arbitrary generators, arbitrary loads	2-level trees	<p>NP-hard</p> <p>[Lehmann et al. 2014]</p>	
$ V_G =2, V_D =2$	planar graphs with max degree of 3	<p>strongly NP-hard</p> <p>[Lehmann et al. 2014]</p>	
	arbitrary graphs	<p>non-APX</p> <p>[Lehmann et al. 2014]</p>	

Subset Sum Problem (SSP)

Decision Problem Subset Sum (SSP)

Instance: A finite set of numbers $W = \{w_1, w_2, \dots, w_n\}$ with $w_i \in \mathbb{N}$ and $k \in \mathbb{N}$.

Question: Is there a set of elements $x_1, x_2, \dots, x_n \in \{0, 1\}$ such that $\sum_{j=1}^n w_j x_j = k$?

Subset Sum Problem (SSP)

Decision Problem Subset Sum (SSP)

Instance: A finite set of numbers $W = \{w_1, w_2, \dots, w_n\}$ with $w_i \in \mathbb{N}$ and $k \in \mathbb{N}$.

Question: Is there a set of elements $x_1, x_2, \dots, x_n \in \{0, 1\}$ such that $\sum_{j=1}^n w_j x_j = k$?

Example Instance

- $W = \{1, 2, 3, 7, 37, 99\}$
- $k = 42$

Subset Sum Problem (SSP)

Decision Problem Subset Sum (SSP)

Instance: A finite set of numbers $W = \{w_1, w_2, \dots, w_n\}$ with $w_i \in \mathbb{N}$ and $k \in \mathbb{N}$.

Question: Is there a set of elements $x_1, x_2, \dots, x_n \in \{0, 1\}$ such that $\sum_{j=1}^n w_j x_j = k$?

Example Instance

- $W = \{1, 2, 3, 7, 37, 99\}$
- $k = 42$

Solution

- $X = \{2, 3, 37\}$

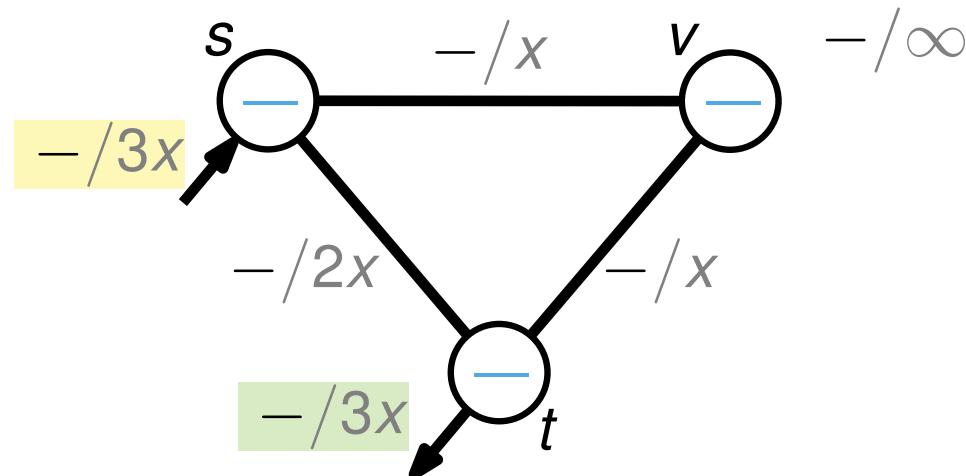
MTSF on Cacti is NP-hard

[Lehmann et al., 2014]

- MTSF for cacti networks with a maximum degree of 3 is NP-hard
- Reduction from Subset Sum
- Switching choice network (SCN) is a gadget encoding decisions representing a network

$$\text{SCN}_{\ell,v} = (\{s, t, v\}, E, \underline{d} := 3\ell, \bar{d} := 3\ell, \bar{x} := 3\ell, \text{cap})$$

Switching choice network SCN



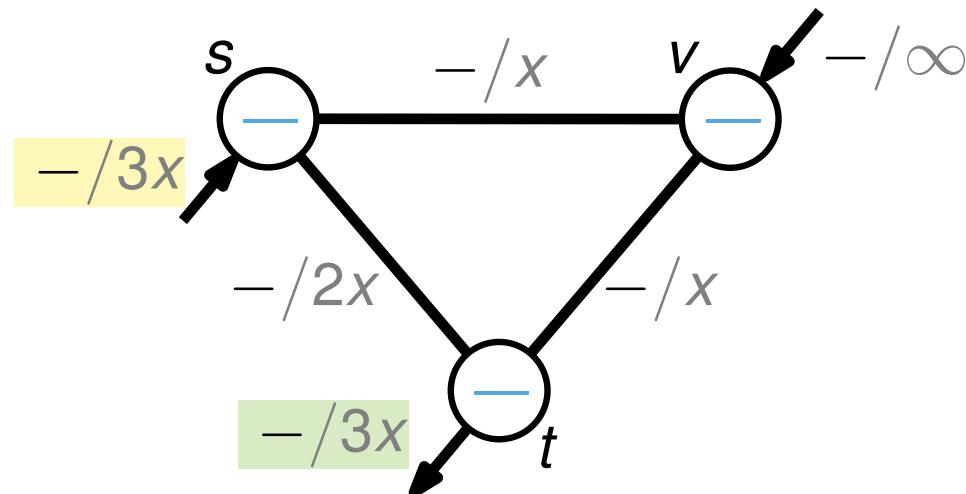
MTSF on Cacti is NP-hard

[Lehmann et al., 2014]

- MTSF for cacti networks with a maximum degree of 3 is NP-hard
- Reduction from Subset Sum
- Switching choice network (SCN) is a gadget encoding decisions representing a network

$$\text{SCN}_{\ell,v} = (\{s, t, v\}, E, \underline{d} := 3\ell, \bar{d} := 3\ell, \bar{x} := 3\ell, \text{cap})$$

Switching choice network SCN_{x,v}⁻



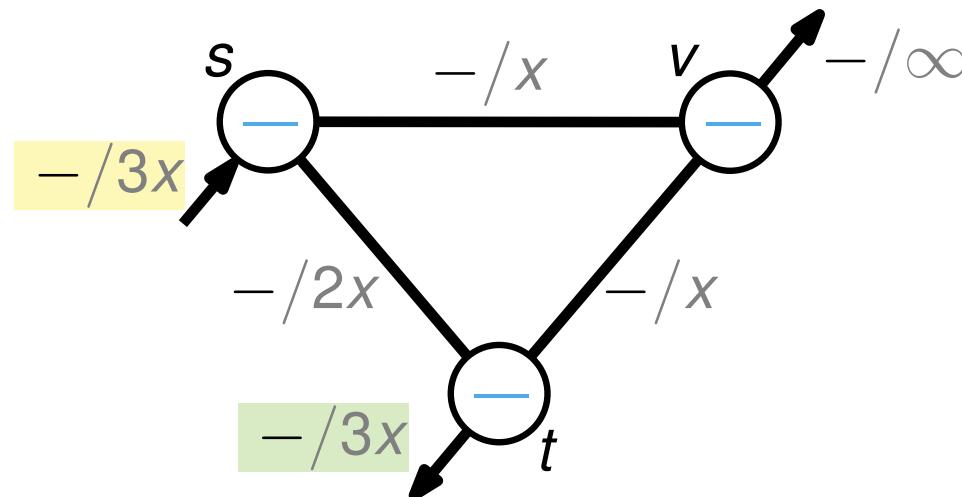
MTSF on Cacti is NP-hard

[Lehmann et al., 2014]

- MTSF for cacti networks with a maximum degree of 3 is NP-hard
- Reduction from Subset Sum
- Switching choice network (SCN) is a gadget encoding decisions representing a network

$$\text{SCN}_{\ell,v} = (\{s, t, v\}, E, \underline{d} := 3\ell, \bar{d} := 3\ell, \bar{x} := 3\ell, \text{cap})$$

Switching choice network $\text{SCN}_{x,v}^+$



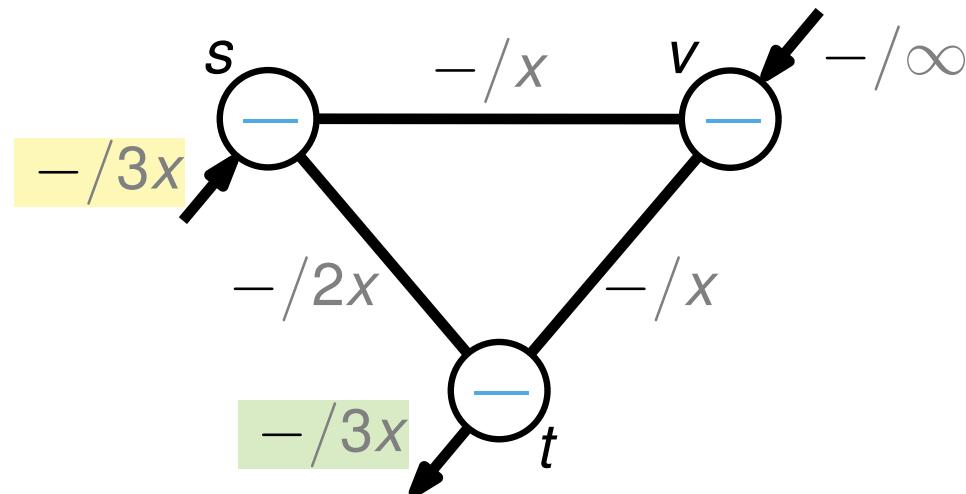
MTSF on Cacti is NP-hard

[Lehmann et al., 2014]

- MTSF for cacti networks with a maximum degree of 3 is NP-hard
- Reduction from Subset Sum
- Switching choice network (SCN) is a gadget encoding decisions representing a network

$$\text{SCN}_{\ell,v} = (\{s, t, v\}, E, \underline{d} := 3\ell, \bar{d} := 3\ell, \bar{x} := 3\ell, \text{cap})$$

Switching choice network SCN_{x,v}⁻



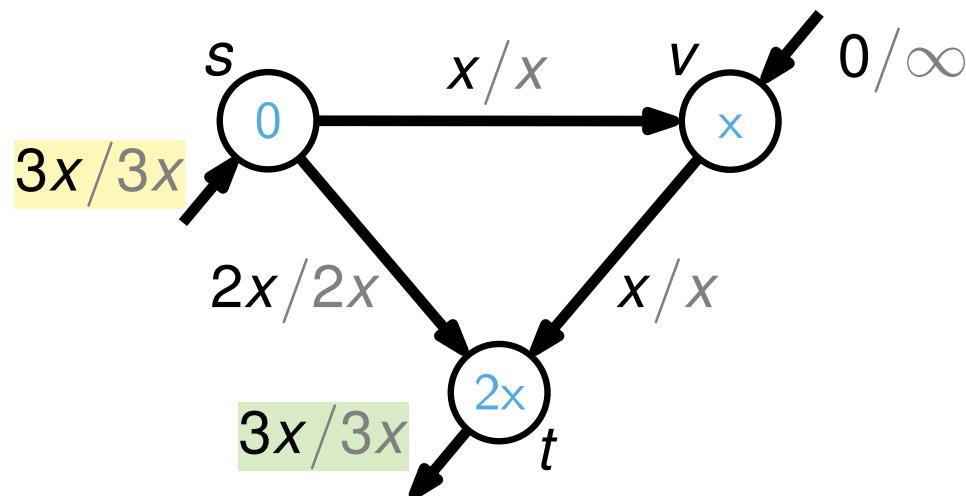
MTSF on Cacti is NP-hard

[Lehmann et al., 2014]

- MTSF for cacti networks with a maximum degree of 3 is NP-hard
- Reduction from Subset Sum
- Switching choice network (SCN) is a gadget encoding decisions representing a network

$$\text{SCN}_{\ell,v} = (\{s, t, v\}, E, \underline{d} := 3\ell, \bar{d} := 3\ell, \bar{x} := 3\ell, \text{cap})$$

Switching choice network $\text{SCN}_{x,v}^-$



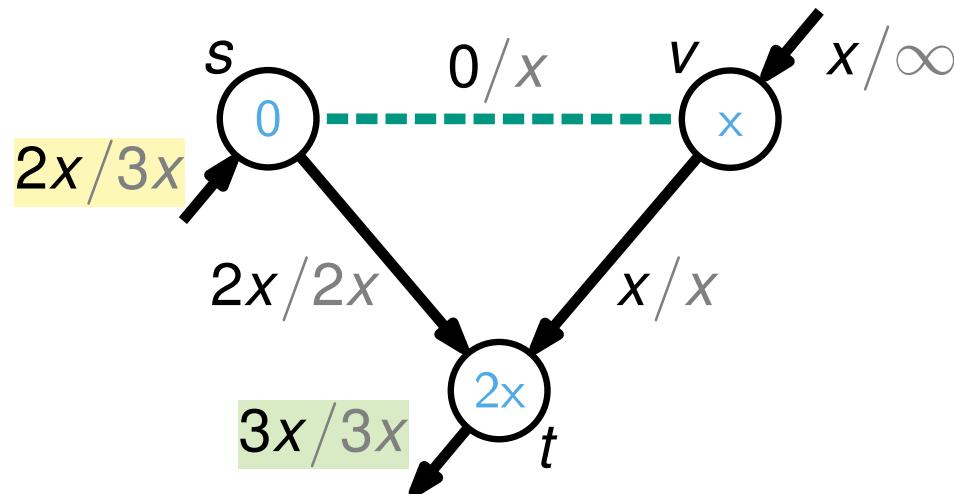
MTSF on Cacti is NP-hard

[Lehmann et al., 2014]

- MTSF for cacti networks with a maximum degree of 3 is NP-hard
- Reduction from Subset Sum
- Switching choice network (SCN) is a gadget encoding decisions representing a network

$$\text{SCN}_{\ell,v} = (\{s, t, v\}, E, \underline{d} := 3\ell, \bar{d} := 3\ell, \bar{x} := 3\ell, \text{cap})$$

Switching choice network $\text{SCN}_{x,v}^-$



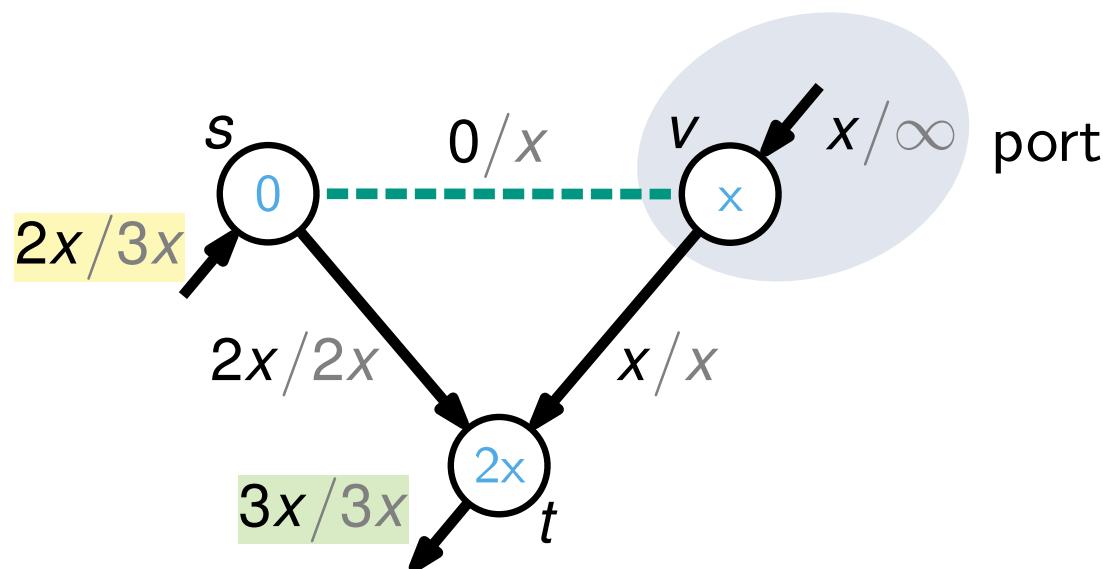
MTSF on Cacti is NP-hard

[Lehmann et al., 2014]

- MTSF for cacti networks with a maximum degree of 3 is NP-hard
- Reduction from Subset Sum
- Switching choice network (SCN) is a gadget encoding decisions representing a network

$$\text{SCN}_{\ell,v} = (\{s, t, v\}, E, \underline{d} := 3\ell, \bar{d} := 3\ell, \bar{x} := 3\ell, \text{cap})$$

Switching choice network $\text{SCN}_{x,v}^-$



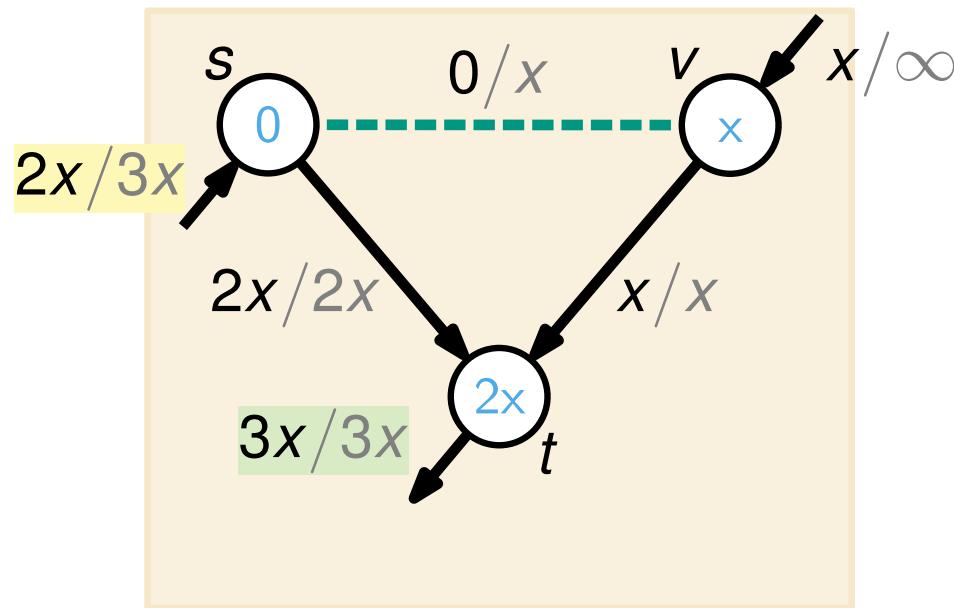
MTSF on Cacti is NP-hard

[Lehmann et al., 2014]

- MTSF for cacti networks with a maximum degree of 3 is NP-hard
- Reduction from Subset Sum
- Switching choice network (SCN) is a gadget encoding decisions representing a network

$$\text{SCN}_{\ell,v} = (\{s, t, v\}, E, \underline{d} := 3\ell, \bar{d} := 3\ell, \bar{x} := 3\ell, \text{cap})$$

Switching choice network $\text{SCN}_{x,v}^-$



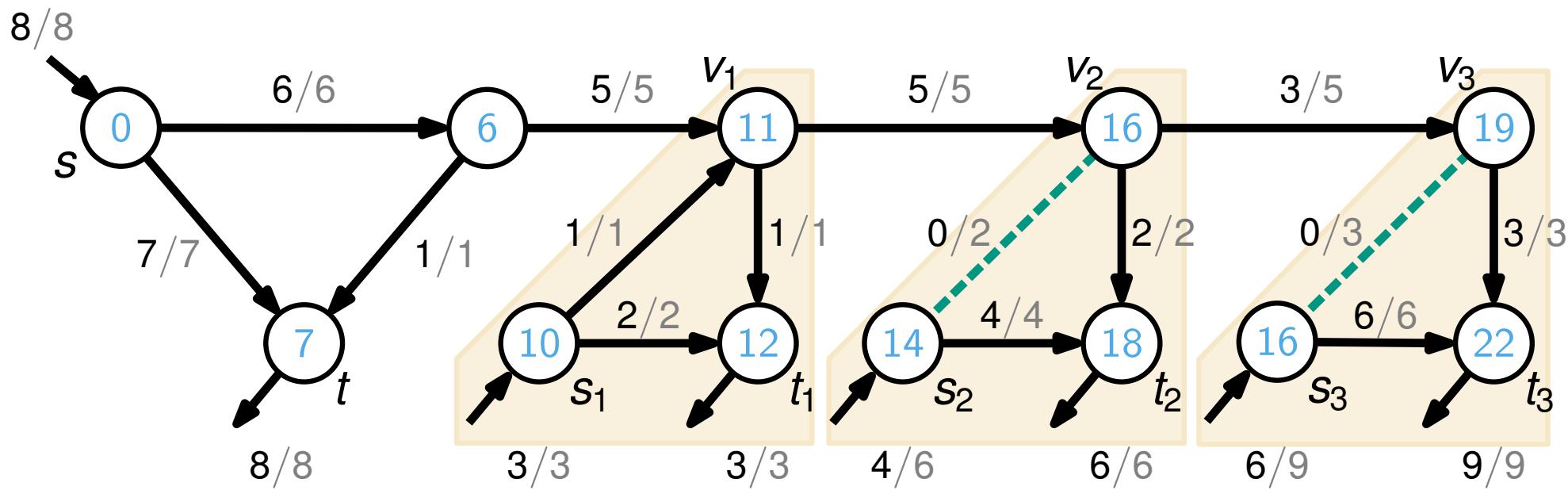
MTSF on Cacti is NP-hard

[Lehmann et al., 2014]

- MTSF for cacti networks with a maximum degree of 3 is NP-hard
- Reduction from Subset Sum
- Switching choice network (SCN) is a gadget encoding decisions representing a network

$$\text{SCN}_{\ell,v} = (\{s, t, v\}, E, \underline{d} := 3\ell, \bar{d} := 3\ell, \bar{x} := 3\ell, \text{cap})$$

Transformation $W = \{1, 2, 3\}$, $k = 5$



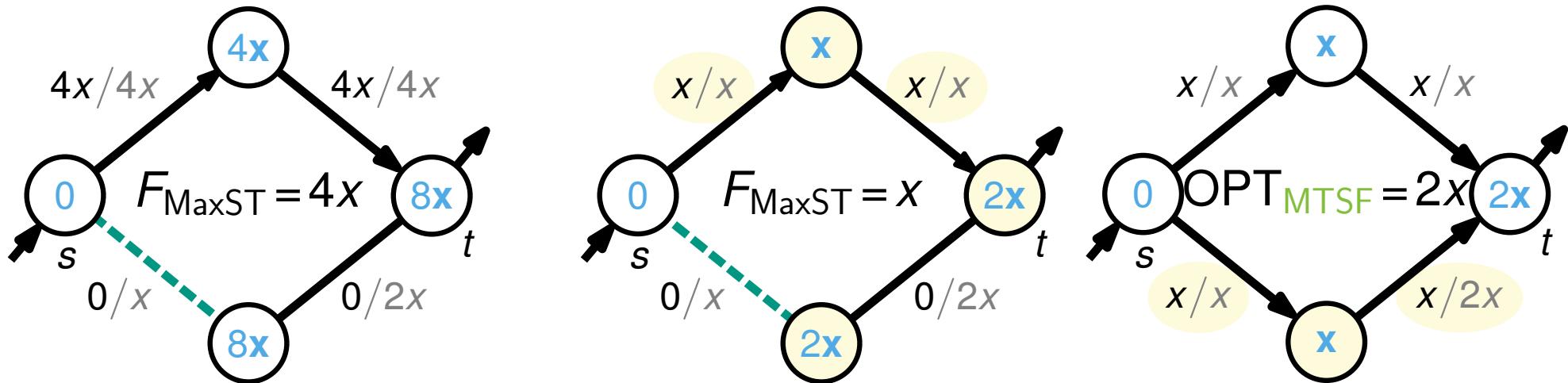
2-approximation on Cacti

Description

- Remove from each cycle the edge with the smallest capacity
 \Leftrightarrow the MAXIMUM SPANNING TREE (MaxST)

MaxST on Cacti

- MTSF is NP-hard on cacti [Lehmann et al., 2014]



Theorem 6 [page 348; Grastien et al., 2018]

MaxST is a factor 2-approximation algorithm for the MF and MTSF problem on cacti.

2-approximation on Cacti

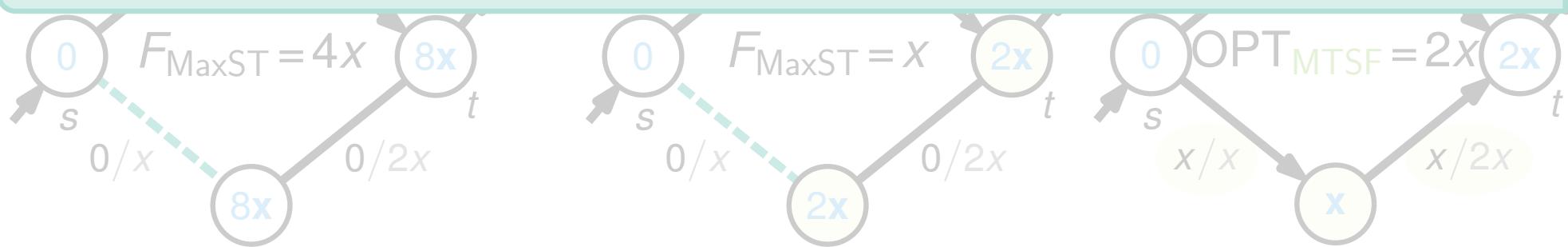
Description

- Remove from each cycle the edge with the smallest capacity
 \Leftrightarrow the MAXIMUM SPANNING TREE (MaxST)

MaxST on Cacti

- MTSF is NP-hard on cacti [Lehmann et al., 2014]

On cacti the MaxST algorithm runs in time $\mathcal{O}(|V|)$.



Theorem 6 [page 348; Grastien et al., 2018]

MaxST is a factor 2-approximation algorithm for the MF and MTSF problem on cacti.

2-approximation on Cacti

Data: A network $\mathcal{N} = (G, V_G, V_D, \text{cap}, \textcolor{brown}{b})$.

Result: $\text{OPT}_{\text{MPF}}(\mathcal{N} - \textcolor{teal}{S})$, and switched edges $\textcolor{teal}{S}$.

```
1  $\textcolor{teal}{S} = \emptyset$ ;  
2  $C = \text{dfs}(\mathcal{N})$ ;  
3 for  $c \in C$  do  
4    $\textcolor{teal}{S} = \textcolor{teal}{S} \cup \{\arg \min_{\forall e \in c} (\text{cap}(e))\}$ ;  
5 end  
6 return  $(\text{OPT}_{\text{MF}}(\mathcal{N} - \textcolor{teal}{S}), \textcolor{teal}{S})$ ;
```

Lemma 7 [page 347; Grastien et al., 2018]

Let $\mathcal{N} = (G, V_G, V_D, \text{cap}, \textcolor{brown}{b})$ be a power grid and let S be the set $\arg \min_{\forall e \in c} \text{cap}(e)$ of switched edges for all cycles $c \in C$. Then there exist an electrically feasible flow f' on $\mathcal{N} - \textcolor{teal}{S}$ such that $F(f') = \frac{1}{2} \text{OPT}_{\text{MF}}(\mathcal{N})$.

PROOF.

$$|f(e)| = \left| \frac{1}{2} f^*(e) \right| \leq \frac{1}{2} \text{cap}(e),$$

$$|f(e_{\min})| \leq \frac{1}{2} \text{cap}(e_{\min}) \leq \frac{1}{2} \text{cap}(e),$$

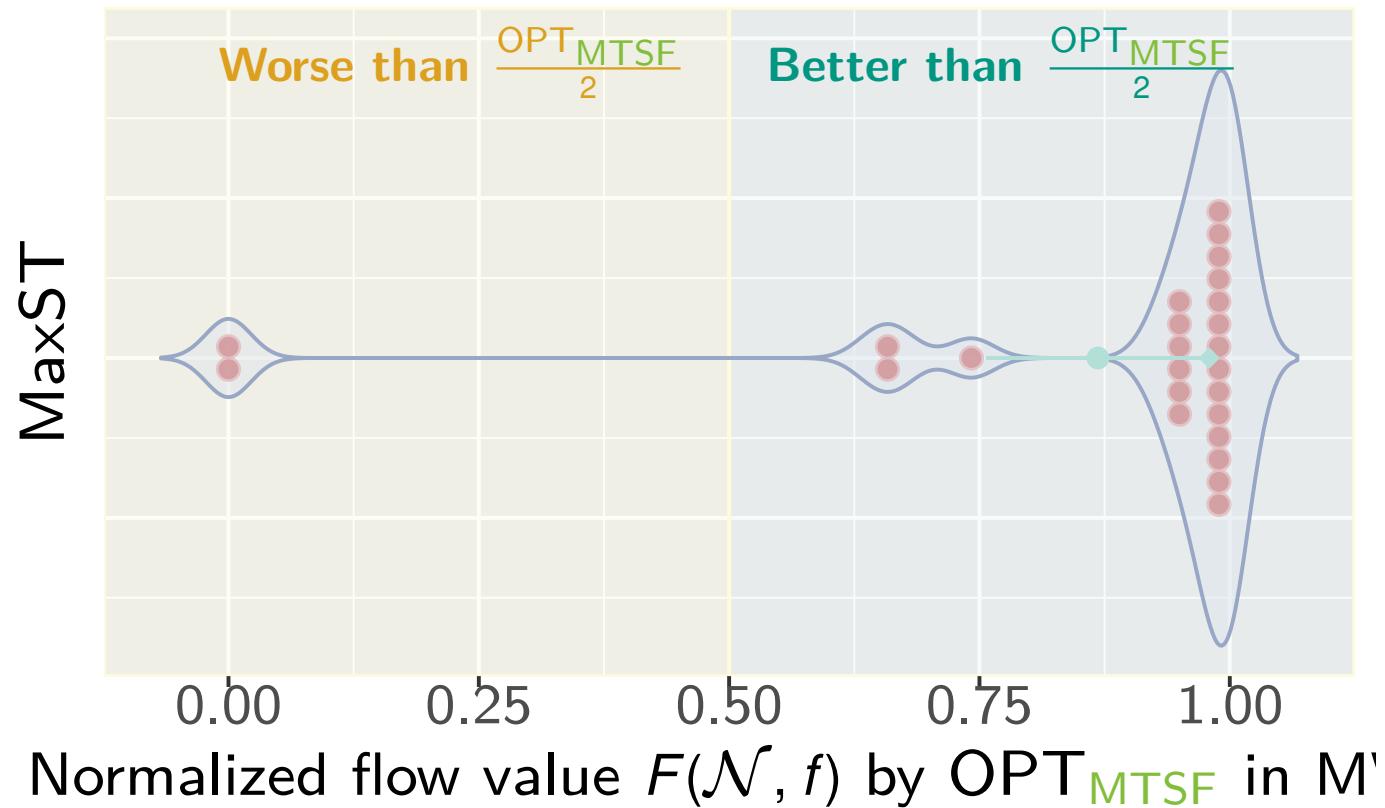
$$|f'(e)| = |f(e_{\min}) + f(e)| \leq \text{cap}(e).$$

□

Simulations

[page 349; Grastien et al., 2018]

- Simulations on NESTA benchmark sets that are more realistic than the IEEE benchmark sets, e.g., with regards to thermal line limits

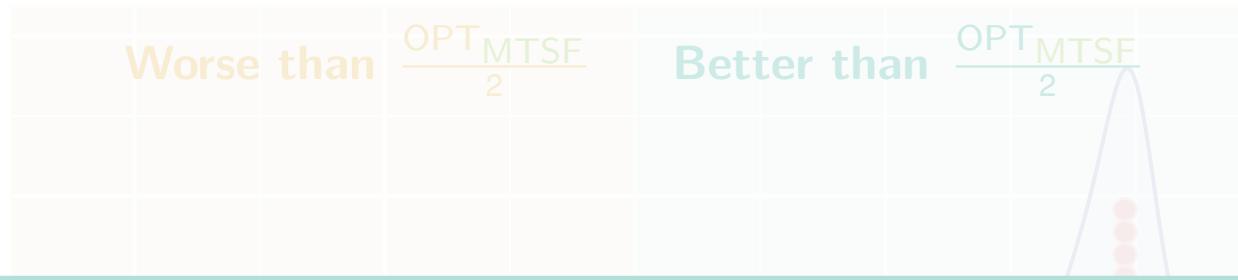


MaxST on **general graphs** is in most cases very close to the OPT_{MTSF} .

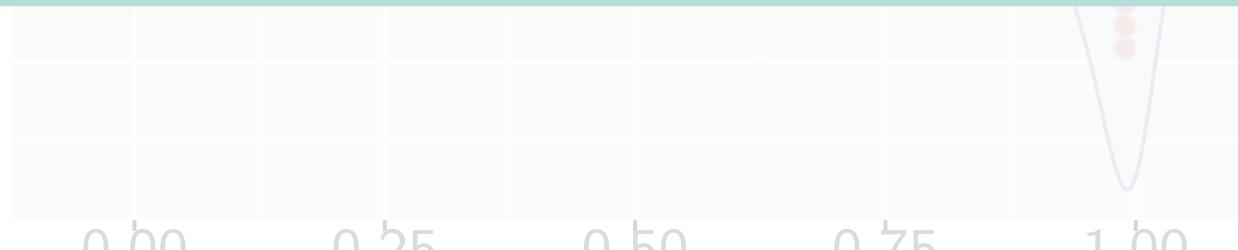
Simulations

[page 349; Grastien et al., 2018]

- Simulations on NESTA benchmark sets that are more realistic than the IEEE benchmark sets, e.g., with regards to thermal line limits



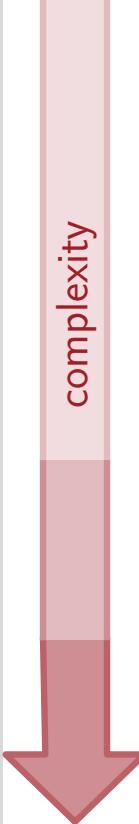
On general graphs the MaxST algorithm runs in time $\mathcal{O}(|E| \alpha(|E|, |V|))$.



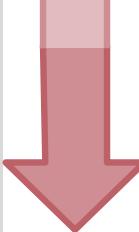
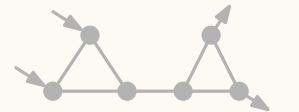
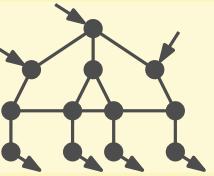
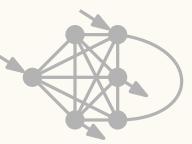
Normalized flow value $F(\mathcal{N}, f)$ by OPT_{MTSF} in M'

MaxST on **general graphs** is in most cases very close to the OPT_{MTSF} .

Overview of the MTSF Results

	Graph Structure	Complexity	Algorithm
complexity 	penrose-minor-free graphs	polynomial-time solvable	DTP 
	series-parallel graphs	NP-hard	
	cacti with max degree of 3	NP-hard [Lehmann et al., 2014]	2-approx. 
	2-level trees	NP-hard [Lehmann et al., 2014]	
	planar graphs with max degree of 3	strongly NP-hard [Lehmann et al., 2014]	
	arbitrary graphs	non-APX [Lehmann et al., 2014]	

Overview of the MTSF Results

	Graph Structure	Complexity	Algorithm
complexity 	penrose-minor-free graphs 	polynomial-time solvable	DTP 
	series-parallel graphs 	NP-hard	
	cacti with max degree of 3 	NP-hard [Lehmann et al., 2014]	2-approx. 
	2-level trees 	NP-hard [Lehmann et al., 2014]	
	planar graphs with max degree of 3 	strongly NP-hard [Lehmann et al., 2014]	
$ V_G = 2, V_D = 2$	arbitrary graphs 	non-APX [Lehmann et al., 2014]	

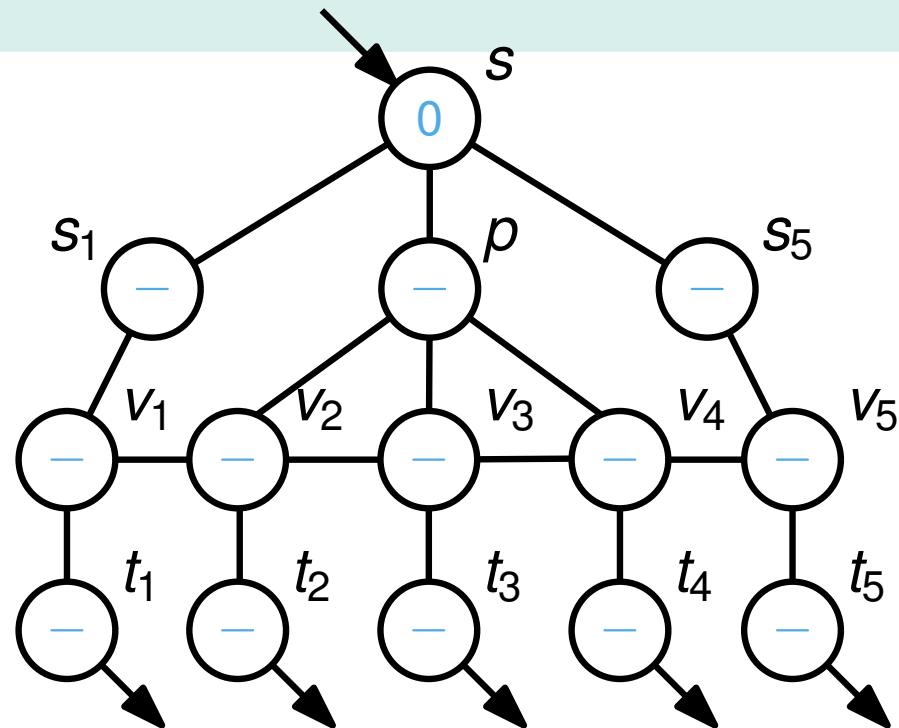
n-level Tree Graphs

[Lehmann et al., 2014]

Definition 8 [page 8; Lehmann et al., 2014]

Let $n \in \mathbb{N}$. A graph G is an n-level tree if and only if there exists a sub-graph T that is a tree such that all leaves of T are sinks $t \in V_D$, and there is only one source $V_G := \{s\}$ at the root vertex of T and there is a total order on the children of every vertex (which implies a total order on all vertices in one level) such that every vertex of the same tree level can only be connected to its neighbours in the total order on all vertices of the same level.

- Possible relaxation of a tree (allowing cycles)
- One generator at the root (super source s) and consumers at the leaves
- Lines not on the tree can only be added between buses on the same tree level
- Planar graph
- Level is less or equal n



n-level Tree Graphs

[Lehmann et al., 2014]

Definition 8 [page 8; Lehmann et al., 2014]

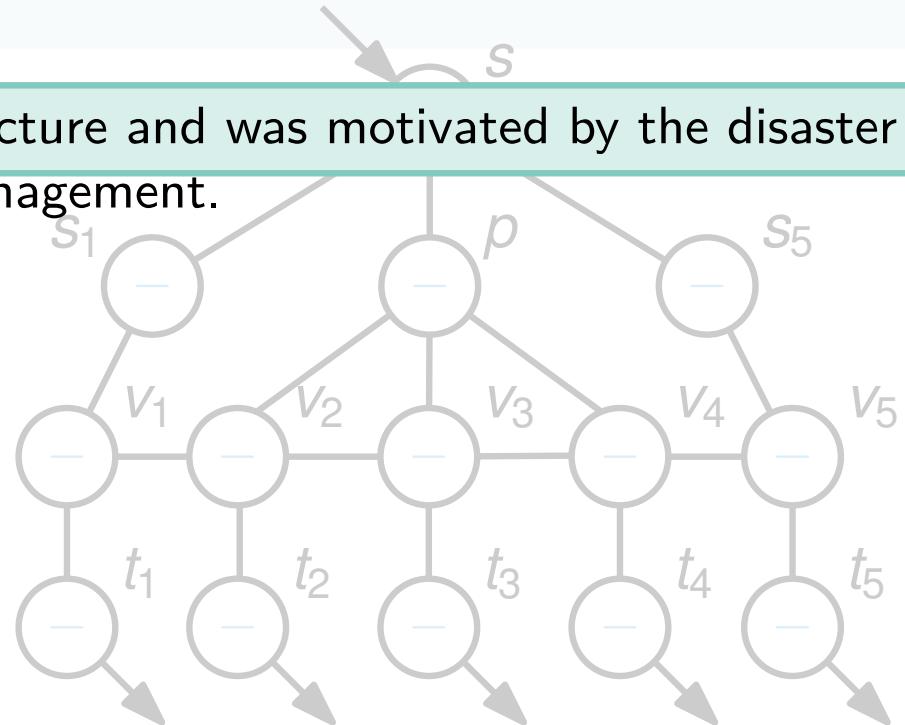
Let $n \in \mathbb{N}$. A graph G is an n-level tree if and only if there exists a sub-graph T that is a tree such that all leaves of T are sinks $t \in V_D$, and there is only one source $V_G := \{s\}$ at the root vertex of T and there is a total order on the children of every vertex (which implies a total order on all vertices in one level) such that every vertex of the same tree level can only be connected to its neighbours in the total order on all vertices of the same level.

Possible relaxation of a tree

This is **not** a standard graph structure and was motivated by the disaster

management.

- One generator at the root (super source s) and consumers at the leaves
- Lines not on the tree can only be added between buses on the same tree level
- Planar graph
- Level is less or equal n



Subset Sum Problem (SSP)

Decision Problem Subset Sum (SSP)

Instance: A finite set of numbers $W = \{w_1, w_2, \dots, w_n\}$ with $w_i \in \mathbb{N}$ and $k \in \mathbb{N}$.

Question: Is there a set of elements $x_1, x_2, \dots, x_n \in \{0, 1\}$ such that $\sum_{j=1}^n w_j x_j = k$?

Subset Sum Problem (SSP)

Decision Problem Subset Sum (SSP)

Instance: A finite set of numbers $W = \{w_1, w_2, \dots, w_n\}$ with $w_i \in \mathbb{N}$ and $k \in \mathbb{N}$.

Question: Is there a set of elements $x_1, x_2, \dots, x_n \in \{0, 1\}$ such that $\sum_{j=1}^n w_j x_j = k$?

Example Instance

- $W = \{1, 2, 3, 7, 37, 99\}$
- $k = 42$

Subset Sum Problem (SSP)

Decision Problem Subset Sum (SSP)

Instance: A finite set of numbers $W = \{w_1, w_2, \dots, w_n\}$ with $w_i \in \mathbb{N}$ and $k \in \mathbb{N}$.

Question: Is there a set of elements $x_1, x_2, \dots, x_n \in \{0, 1\}$ such that $\sum_{j=1}^n w_j x_j = k$?

Example Instance

- $W = \{1, 2, 3, 7, 37, 99\}$
- $k = 42$

Solution

- $X = \{2, 3, 37\}$

MTSF on 2-level Trees

[Lehmann et al., 2014]

Theorem 9 [page 9; Lehmann et al., 2014]

The MTSF problem for 2-level trees is NP-hard.

MTSF on 2-level Trees

[Lehmann et al., 2014]

Instance $\mathcal{N} = (G = (V, E), V_G, V_D, \text{cap}, \underline{b}, \bar{x}, \underline{d}, \bar{d})$

- $W = \{w_2, w_3, \dots, w_n\}$, k , $\bar{x} = \infty$, $\underline{d} = m + 2 + k$, and $\bar{d} = m + 2 + \sum_{i=2}^n w_i$
- $n := |W| + 1$
- $m := 1 + \sum_{i=2}^n i \cdot x_i$

vertices ■ $2n + 6$ vertices $\{t_i, v_i \mid 1 \leq i \leq n + 1\} \cup \{s, s_1, s_5, p\}$

edges ■ $3n + 5$ edges
 ■ $(s, s_1), (s, s_{n+1}), (s, p), (s_1, v_1), (s_{n+1}, v_{n+1}),$
 $(p, v_i),$
 $(v_i, t_i),$
 $(v_{i-1}, v_i).$

capacities ■ w_i for $(p, v_i), (v_i, t_i),$
 ■ 1 for $(s, s_{n+1}), (s_{n+1}, v_{n+1}), (v_1, t_1),$
 ■ m for $(v_{i-1}, v_i),$
 ■ $m + 1$ for $(s, s_1), (s_1, v_1), (v_{n+1}, t_{n+1}).$

susceptances ■ $\frac{w_i}{i}$ for $(p, v_i),$
 ■ 1 for $(v_i, t_i),$
 ■ m for $(v_{i-1}, v_i),$
 ■ $2m + 2$ for $(s, s_1), (s_1, v_1),$
 ■ $\frac{2}{n+1}$ for $(s, s_{n+1}), (s_{n+1}, v_{n+1}),$
 ■ k for $(s, p).$

generation / load ■ generation of $m + 2 + k$ at bus s ,
 consumption of $m + 2 + k$ at buses t_i for all $1 \leq i \leq n + 1$

$$\forall i \in \{2, \dots, n\}$$

$$\forall i \in \{1, \dots, n + 1\}$$

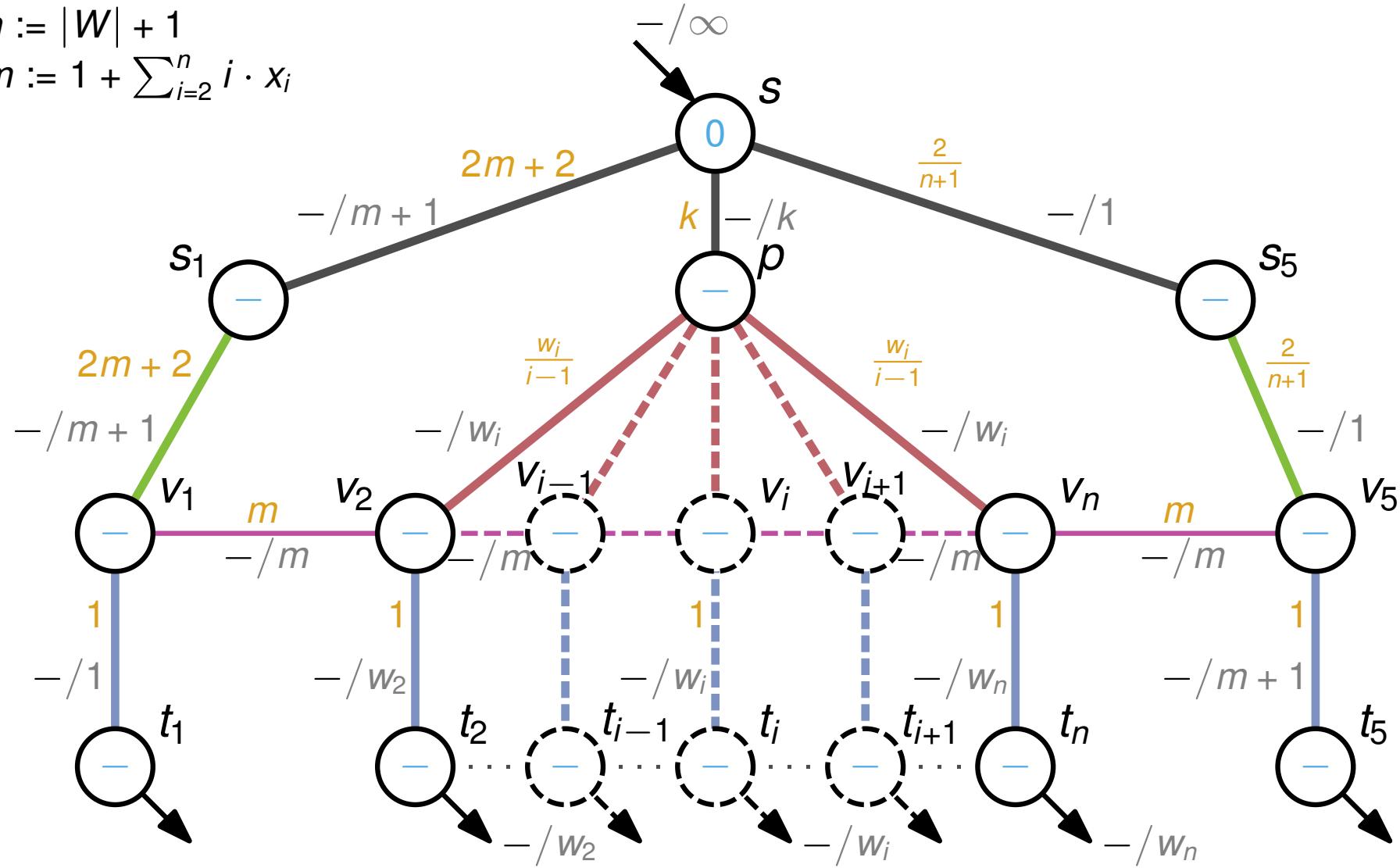
$$\forall i \in \{2, \dots, n + 1\}$$

MTSF on 2-level Trees

[Lehmann et al., 2014]

Instance $\mathcal{N} = (G = (V, E), V_G, V_D, \text{cap}, b, \bar{x}, \underline{d}, \bar{d})$

- $W = \{w_2, w_3, \dots, w_n\}$, k , $\bar{x} = \infty$, $\underline{d} = m + 2 + k$, and $\bar{d} = m + 2 + \sum_{i=2}^n w_i$
 - $n := |W| + 1$
 - $m := 1 + \sum_{i=2}^n i \cdot x_i$

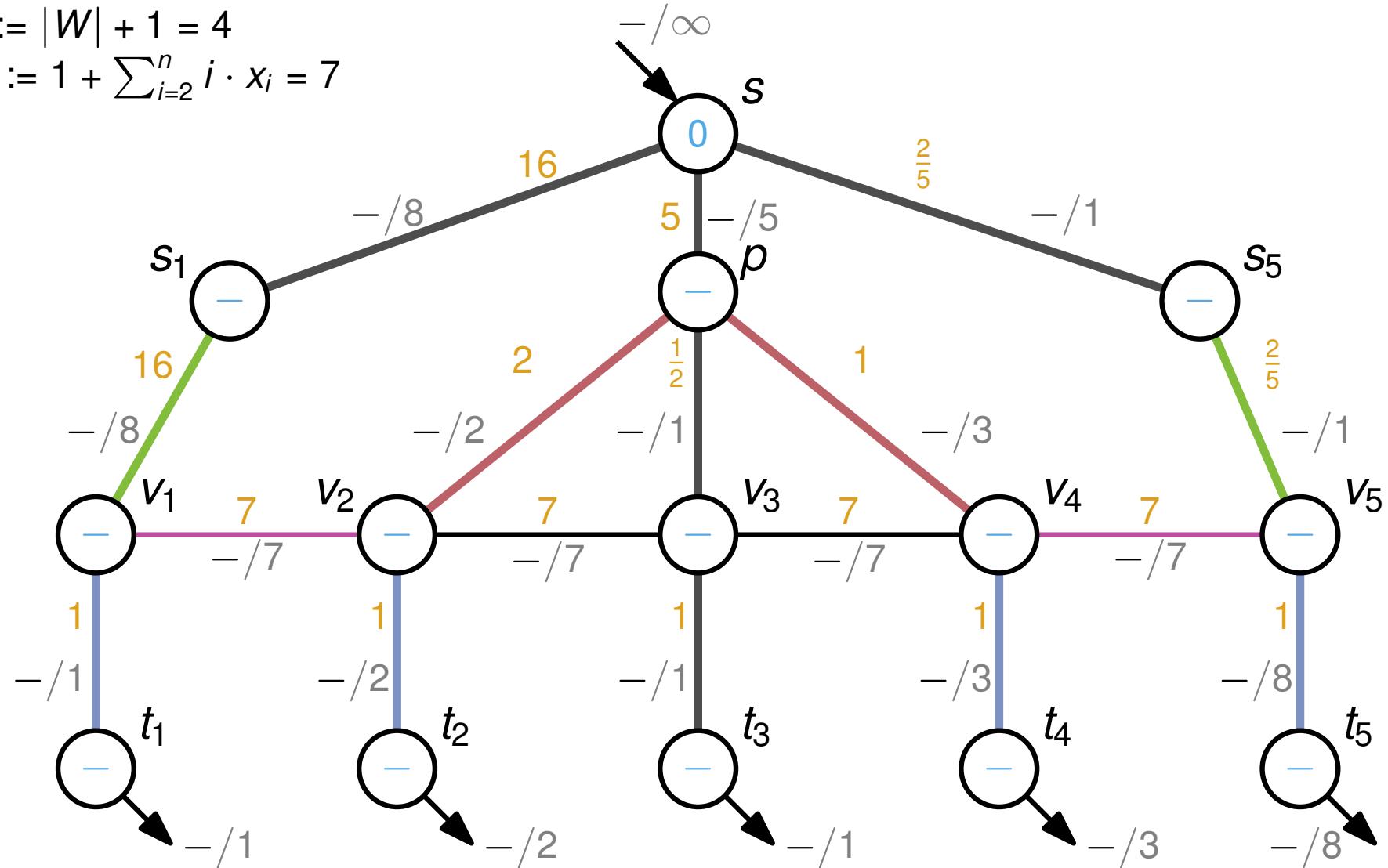


MTSF on 2-level Trees

[Lehmann et al., 2014]

Example Instance $\mathcal{N} = (G = (V, E), V_G, V_D, \text{cap}, \underline{b}, \bar{x}, \underline{d}, \bar{d})$

- $W = \{2, 1, 3\}$, $k = 5$, $\bar{x} = \infty$, $\underline{d} = 14$, and $\bar{d} = 15$
- $n := |W| + 1 = 4$
- $m := 1 + \sum_{i=2}^n i \cdot x_i = 7$

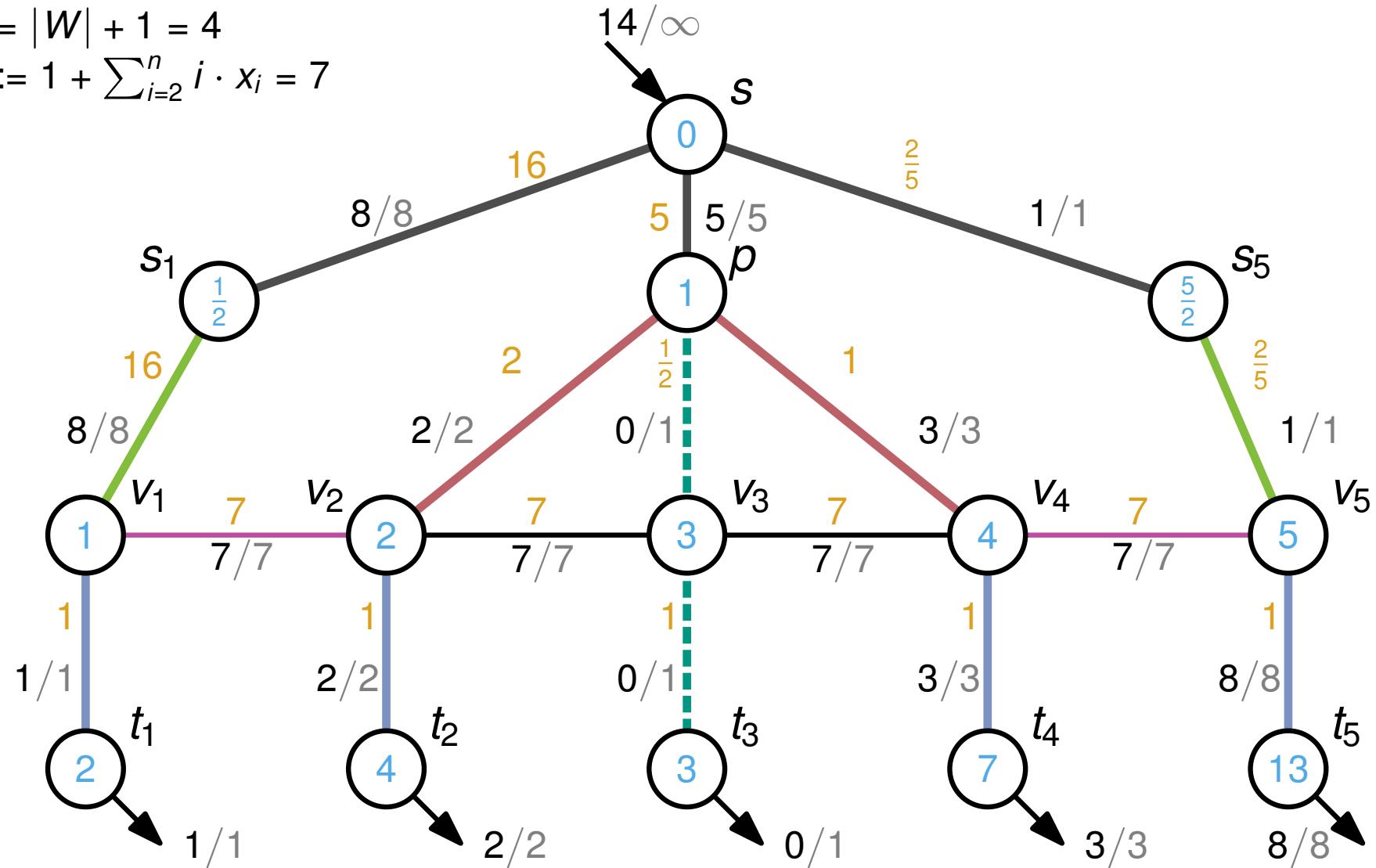


MTSF on 2-level Trees

[Lehmann et al., 2014]

Example Instance $\mathcal{N} = (G = (V, E), V_G, V_D, \text{cap}, \underline{b}, \bar{x}, \underline{d}, \bar{d})$

- $W = \{2, 1, 3\}$, $k = 5$, $\bar{x} = \infty$, $\underline{d} = 14$, and $\bar{d} = 15$
- $n := |W| + 1 = 4$
- $m := 1 + \sum_{i=2}^n i \cdot x_i = 7$

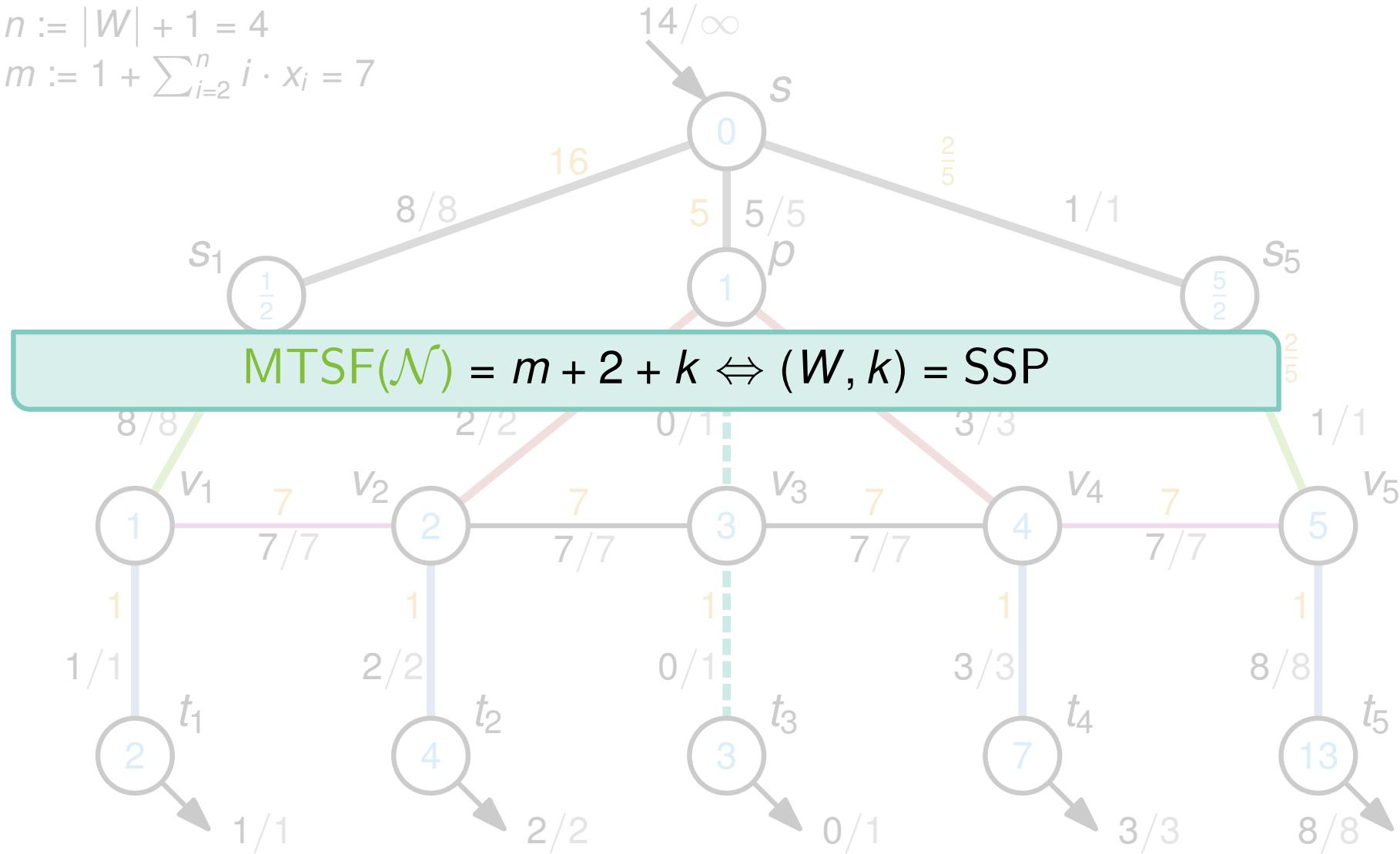


MTSF on 2-level Trees

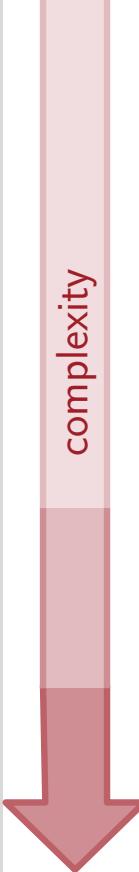
[Lehmann et al., 2014]

Example Instance $\mathcal{N} = (G = (V, E), V_G, V_D, \text{cap}, \underline{b}, \bar{x}, \underline{d}, \bar{d})$

- $W = \{2, 1, 3\}$, $k = 5$, $\bar{x} = \infty$, $\underline{d} = 14$, and $\bar{d} = 15$
- $n := |W| + 1 = 4$
- $m := 1 + \sum_{i=2}^n i \cdot x_i = 7$



Overview of the MTSF Results

	Graph Structure	Complexity	Algorithm
complexity 	penrose-minor-free graphs	polynomial-time solvable	DTP 
	series-parallel graphs	NP-hard	
	cacti with max degree of 3	NP-hard [Lehmann et al., 2014]	2-approx. 
	2-level trees	NP-hard [Lehmann et al., 2014]	
	planar graphs with max degree of 3	strongly NP-hard [Lehmann et al., 2014]	
	arbitrary graphs	non-APX [Lehmann et al., 2014]	
	$ V_G = 2, V_D = 2$		

Overview of the MTSF Results

	Graph Structure	Complexity	Algorithm
complexity ↓	penrose-minor-free graphs	polynomial-time solvable	DTP
	series-parallel graphs	NP-hard	
	cacti with max degree of 3	NP-hard [Lehmann et al., 2014]	2-approx.
	2-level trees	NP-hard [Lehmann et al., 2014]	
	planar graphs with max degree of 3	strongly NP-hard [Lehmann et al., 2014]	
	arbitrary graphs	non-APX [Lehmann et al., 2014]	

Hamilton Path Problem (HPP) & Hamilton Cycle Problem (HCP)

[page 196; Skiena, 1990]

Hamilton Path Problem (HPP)

[pages 199–200; Garey and Johnson, 1983]

Instance: A graph $G_{\text{HPP}} = (V_{\text{HPP}}, E_{\text{HPP}})$.

Question: Is there a path $\pi^* \in \Pi$ that visits each vertex exactly once.

Hamilton Cycle Problem (HCP)

[Karp, 1972] [page 199; Garey and Johnson, 1983]

Instance: A graph $G_{\text{HCP}} = (V_{\text{HCP}}, E_{\text{HCP}})$.

Question: Is there a graph cycle (i.e., closed loop) in G_{HCP} that visits each vertex exactly once.

- A graph containing a Hamiltonian path is called *traceable graph*
- Hamiltonian cycle is called Hamiltonian graph
- Hamiltonian cycle is also called Hamiltonian circuit
- HCP is strongly NP-complete even for planar and cubic graphs

[pages 95; Garey and Johnson 1983]

MTSF on Planar Graphs is strongly NP-hard

[Lehmann et al., 2014]

Corollary 10 [page 13; Lehmann et al., 2014]

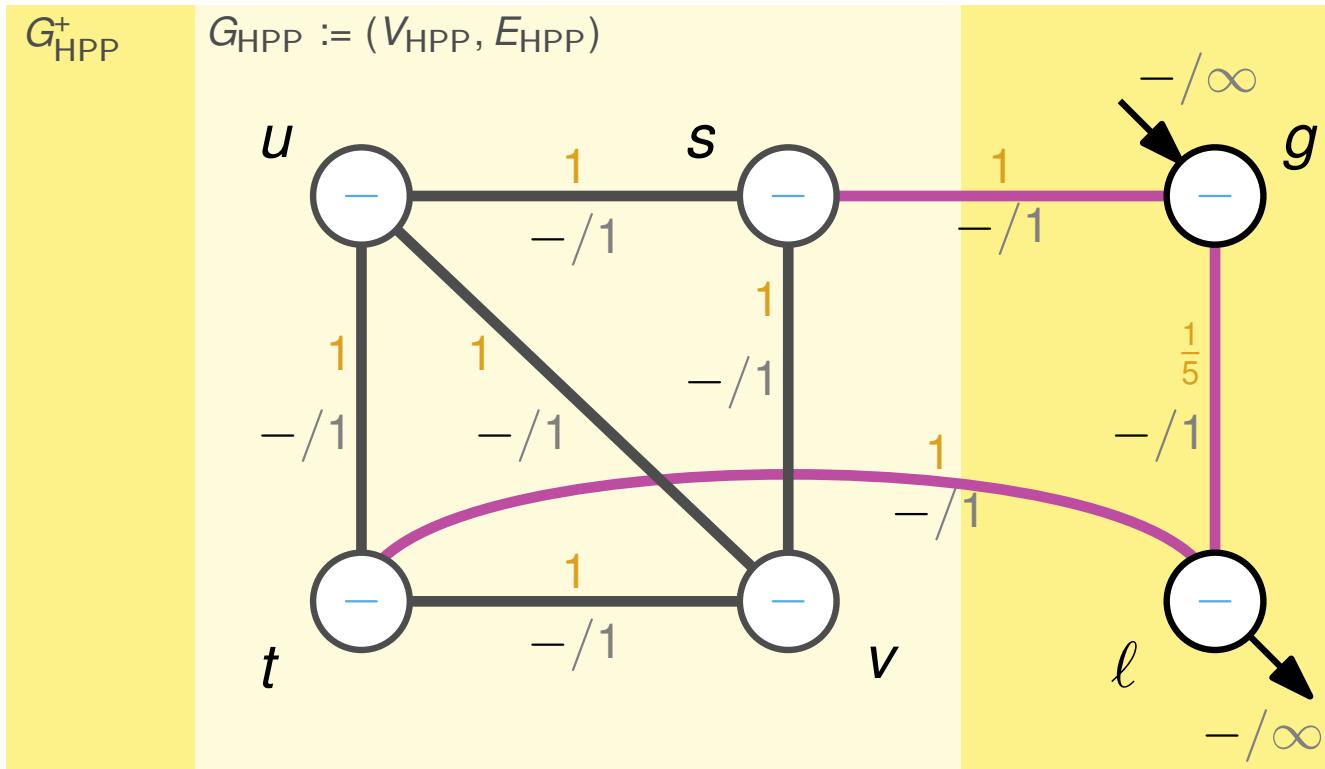
The feasibility problem for planar networks with a maximum degree of 3 is strongly NP-hard.

MTSF on Planar Graphs is strongly NP-hard

[Lehmann et al., 2014]

Instance $\mathcal{N} = (G = (V \cup V_{\text{HPP}}, E \cup E_{\text{HPP}}), V_G = \{g\}, V_D = \{\ell\}, \text{cap}, \mathbf{b}, \bar{x}, \bar{d})$

- Set of vertices $V_{\text{HPP}} = \{v_1, v_2, \dots, v_n\}$, and $\bar{x} = \bar{d} = \infty$,
- Set of edges $E = E_{\text{HPP}} \cup \{(g, s), (t, \ell), (g, \ell)\}$,
- Edge parameters $\text{cap}(e) := \mathbf{b}(e) := 1$ $\forall e \in E \setminus \{(g, \ell)\}$,
 $\text{cap}(g, \ell) := 1, \mathbf{b}(g, \ell) := \frac{1}{n+1},$

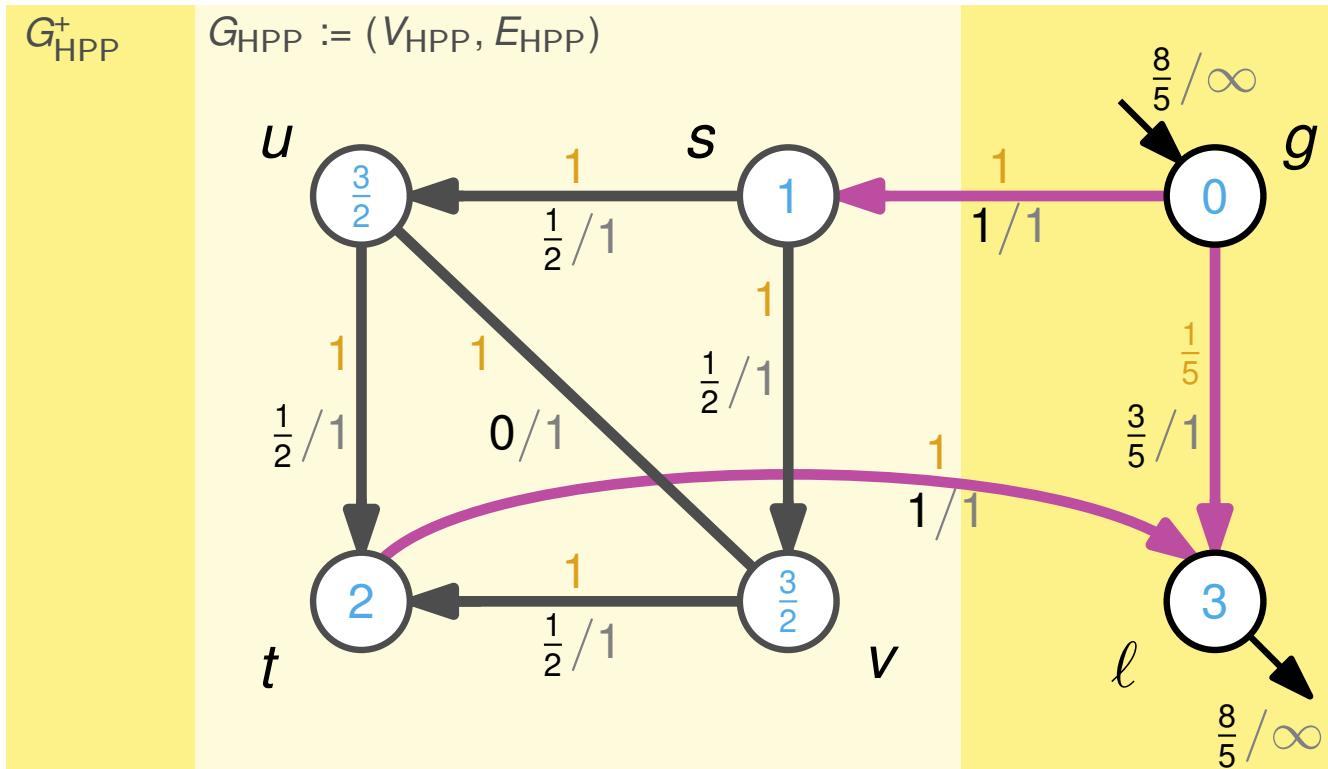


MTSF on Planar Graphs is strongly NP-hard

[Lehmann et al., 2014]

Instance $\mathcal{N} = (G = (V \cup V_{\text{HPP}}, E \cup E_{\text{HPP}}), V_G = \{g\}, V_D = \{\ell\}, \text{cap}, \mathbf{b}, \bar{x}, \bar{d})$

- Set of vertices $V_{\text{HPP}} = \{v_1, v_2, \dots, v_n\}$, and $\bar{x} = \bar{d} = \infty$,
- Set of edges $E = E_{\text{HPP}} \cup \{(g, s), (t, \ell), (g, \ell)\}$,
- Edge parameters $\text{cap}(e) := \mathbf{b}(e) := 1$ $\forall e \in E \setminus \{(g, \ell)\}$,
- $\text{cap}(g, \ell) := 1, \mathbf{b}(g, \ell) := \frac{1}{n+1},$



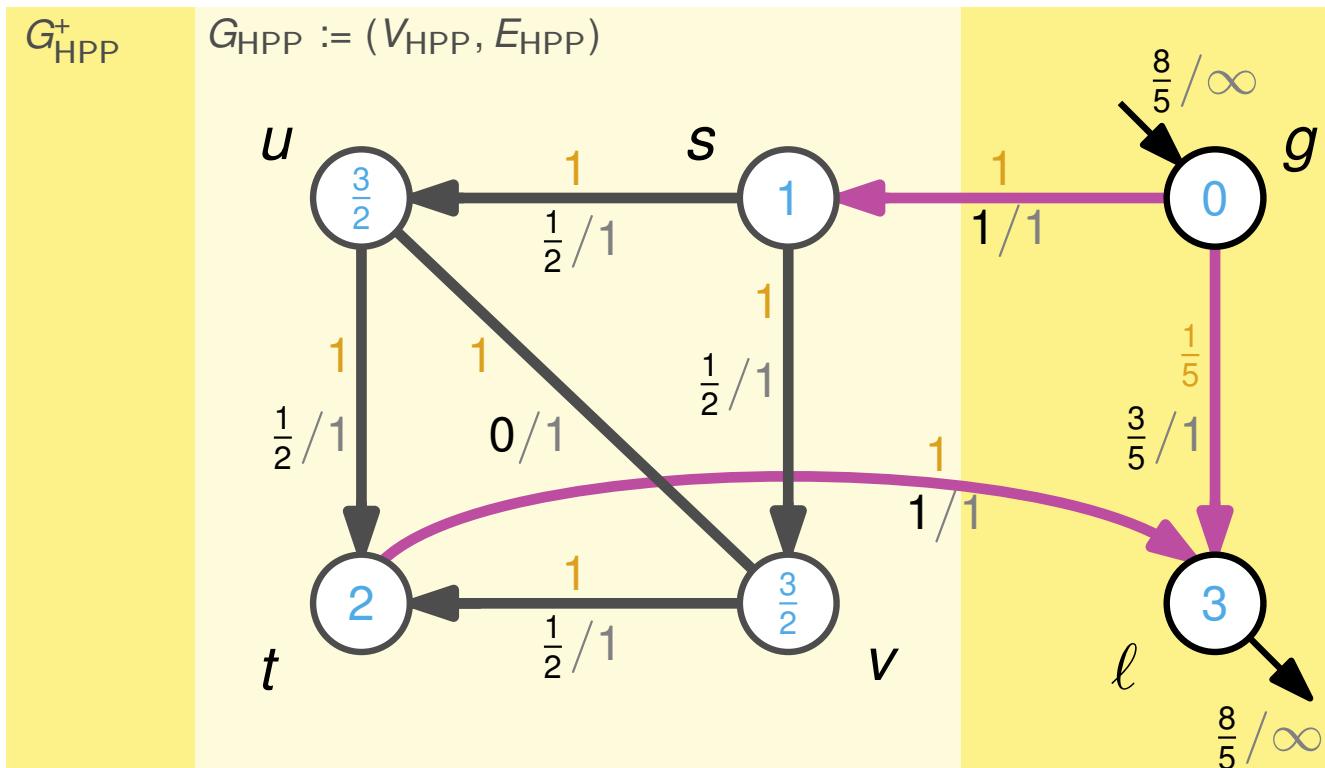
$$\text{OPT}_{\text{MPF}}(\mathcal{N}) = \frac{8}{5}$$

MTSF on Planar Graphs is strongly NP-hard

[Lehmann et al., 2014]

Instance $\mathcal{N} = (G = (V \cup V_{\text{HPP}}, E \cup E_{\text{HPP}}), V_G = \{g\}, V_D = \{\ell\}, \text{cap}, \mathbf{b}, \bar{x}, \bar{d})$

- Set of vertices $V_{\text{HPP}} = \{v_1, v_2, \dots, v_n\}$, and $\bar{x} = \bar{d} = \infty$,
- Set of edges $E = E_{\text{HPP}} \cup \{(g, s), (t, \ell), (g, \ell)\}$,
- Edge parameters $\text{cap}(e) := \mathbf{b}(e) := 1$ $\forall e \in E \setminus \{(g, \ell)\}$,
- $\text{cap}(g, \ell) := 1, \mathbf{b}(g, \ell) := \frac{1}{n+1},$



G_{HPP} limits the $\max \Delta\theta^\nu(s, t)$ that limits the maximum power flow in \mathcal{N}

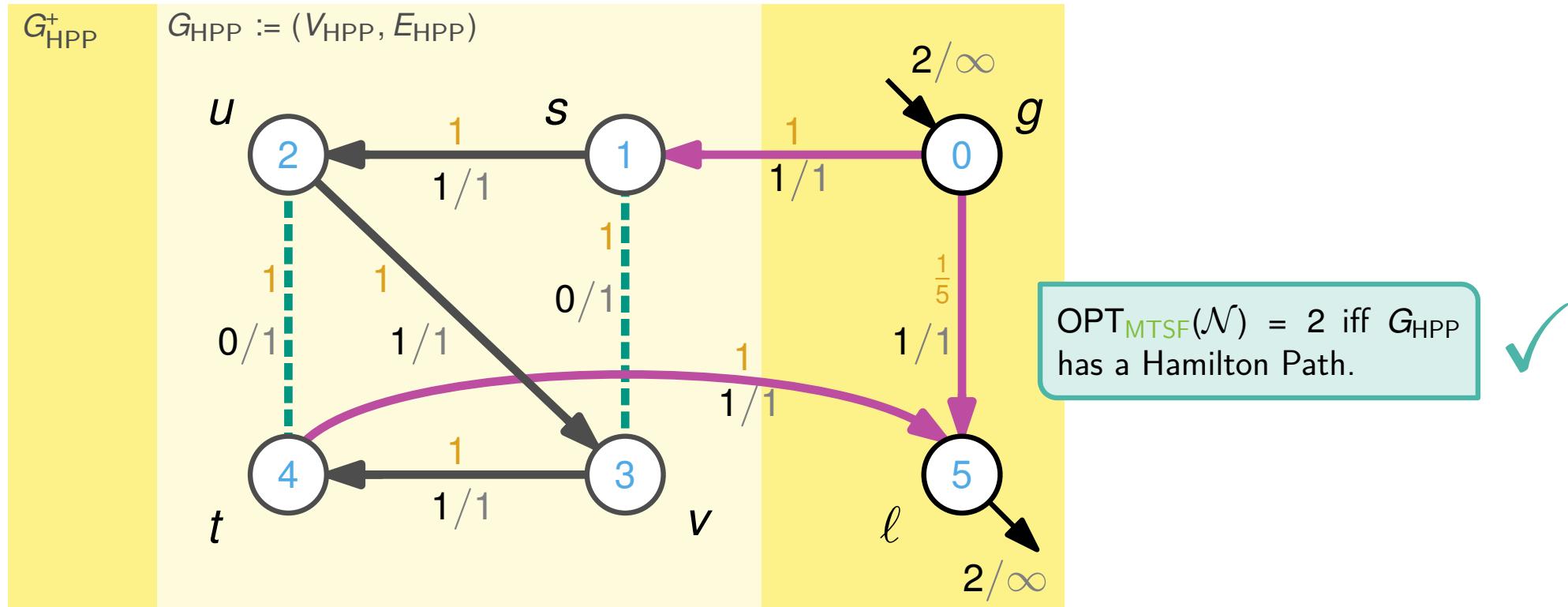
$$\text{OPT}_{\text{MPF}}(G_{\text{HPP}}^+) = \frac{8}{5}$$

MTSF on Planar Graphs is strongly NP-hard

[Lehmann et al., 2014]

Instance $\mathcal{N} = (G = (V \cup V_{\text{HPP}}, E \cup E_{\text{HPP}}), V_G = \{g\}, V_D = \{\ell\}, \text{cap}, \mathbf{b}, \bar{x}, \bar{d})$

- Set of vertices $V_{\text{HPP}} = \{v_1, v_2, \dots, v_n\}$, and $\bar{x} = \bar{d} = \infty$,
- Set of edges $E = E_{\text{HPP}} \cup \{(g, s), (t, \ell), (g, \ell)\}$,
- Edge parameters $\text{cap}(e) := \mathbf{b}(e) := 1$ $\forall e \in E \setminus \{(g, \ell)\}$,
 $\text{cap}(g, \ell) := 1, \mathbf{b}(g, \ell) := \frac{1}{n+1},$



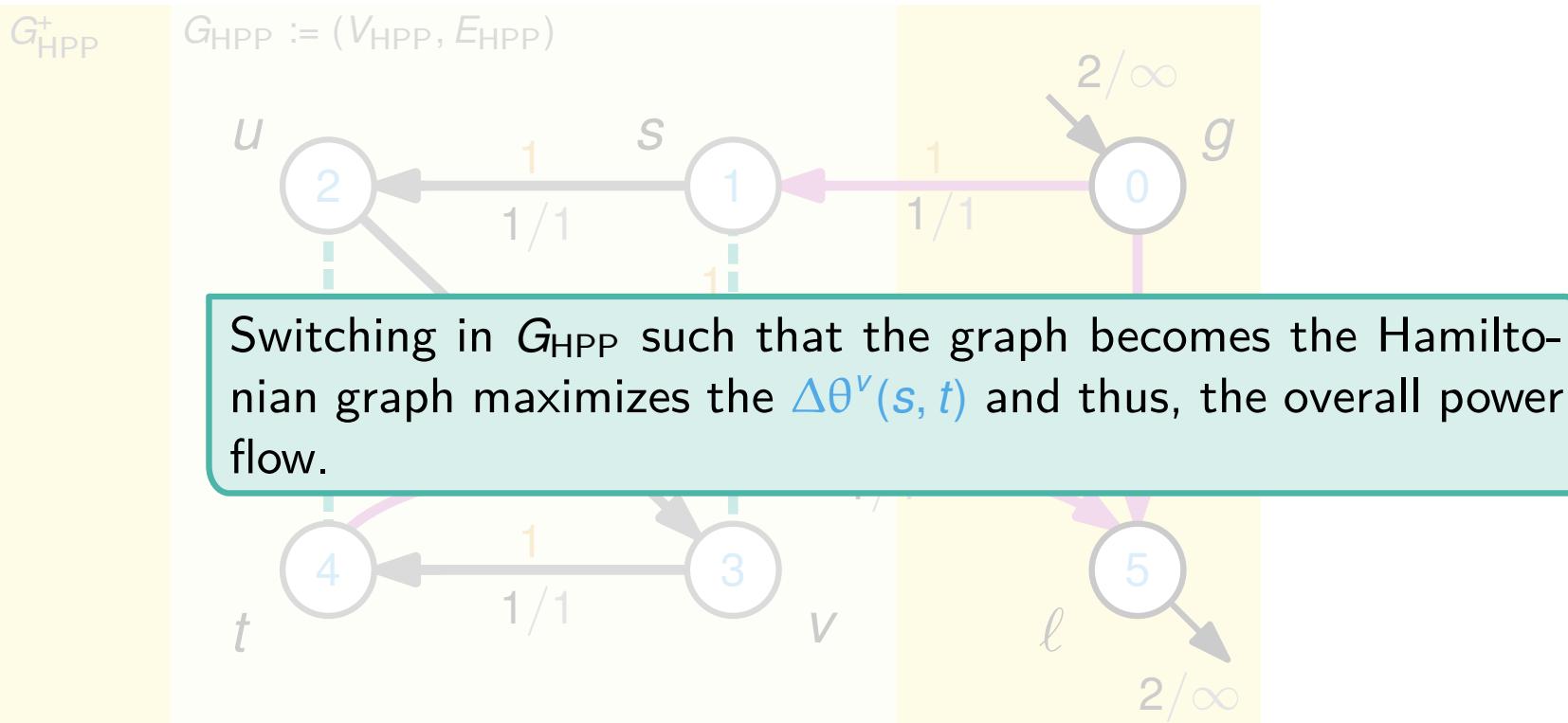
$$\text{OPT}_{\text{MTSF}}(G_{\text{HPP}}^+) = 2$$

MTSF on Planar Graphs is strongly NP-hard

[Lehmann et al., 2014]

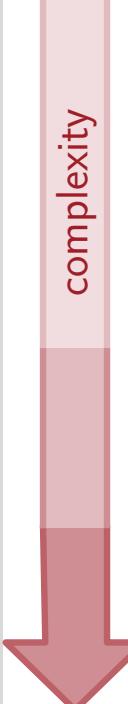
Instance $\mathcal{N} = (G = (V \cup V_{\text{HPP}}, E \cup E_{\text{HPP}}), V_G = \{g\}, V_D = \{\ell\}, \text{cap}, \mathbf{b}, \bar{x}, \bar{d})$

- Set of vertices $V_{\text{HPP}} = \{v_1, v_2, \dots, v_n\}$, and $\bar{x} = \bar{d} = \infty$,
- Set of edges $E = E_{\text{HPP}} \cup \{(g, s), (t, \ell), (g, \ell)\}$,
- Edge parameters $\text{cap}(e) := \mathbf{b}(e) := 1$ $\forall e \in E \setminus \{(g, \ell)\}$,
 $\text{cap}(g, \ell) := 1, \mathbf{b}(g, \ell) := \frac{1}{n+1},$



$$\text{OPT}_{\text{MTSF}}(G_{\text{HPP}}^+) = 2$$

Overview of the MTSF Results

	Graph Structure	Complexity	Algorithm
complexity 	penrose-minor-free graphs	polynomial-time solvable	DTP 
	series-parallel graphs	NP-hard	
	cacti with max degree of 3	NP-hard [Lehmann et al., 2014]	2-approx. 
	2-level trees	NP-hard [Lehmann et al., 2014]	
	planar graphs with max degree of 3	strongly NP-hard [Lehmann et al., 2014]	
	arbitrary graphs	non-APX [Lehmann et al., 2014]	
	$ V_G = 2, V_D = 2$		

Overview of the MTSF Results

	Graph Structure	Complexity	Algorithm
complexity 	penrose-minor-free graphs	polynomial-time solvable	DTP 
	series-parallel graphs	NP-hard	
	cacti with max degree of 3	NP-hard [Lehmann et al., 2014]	2-approx. 
	2-level trees	NP-hard [Lehmann et al., 2014]	
	planar graphs with max degree of 3	strongly NP-hard [Lehmann et al., 2014]	
	arbitrary graphs	non-APX [Lehmann et al., 2014]	

Longest Path (LP)

[Karger et al., 1997]

Longest Path (LP)

Instance: A network $\mathcal{N}_{\text{LP}} = (G_{\text{LP}} = (V_{\text{LP}}, E_{\text{LP}}), \text{len})$, a length function $\text{len}: E \rightarrow \mathbb{R}_{\geq 0}$ and two designated vertices $s, t \in V$.

Question: Find a simple path $\pi^* \in \Pi(s, t)$ from s to t of maximum length—meaning $\max_{\pi \in \Pi(s, t)} \sum_{(u, v) \in \pi} \text{len}(u, v)$ —in the network \mathcal{N}_{LP} .

- Also called TAXICAB RIPOFF problem

Longest Path (LP)

[Karger et al., 1997]

Longest Path (LP)

Instance: A network $\mathcal{N}_{\text{LP}} = (G_{\text{LP}} = (V_{\text{LP}}, E_{\text{LP}}), \text{len})$, a length function $\text{len}: E \rightarrow \mathbb{R}_{\geq 0}$ and two designated vertices $s, t \in V$.

Question: Find a simple path $\pi^* \in \Pi(s, t)$ from s to t of maximum length—meaning $\max_{\pi \in \Pi(s, t)} \sum_{(u, v) \in \pi} \text{len}(u, v)$ —in the network \mathcal{N}_{LP} .

- Also called TAXICAB RIPOFF problem
- For all $\epsilon > 0$ not possible to approximate LP within a factor of $2^{(\log n)^{1-\epsilon}}$ unless NP is contained in quasi-polynomial deterministic time.

MTSF on Arbitrary Graphs is non-APX

[Lehmann et al., 2014]

Theorem 11 [pages 10–12; Lehmann et al., 2014]

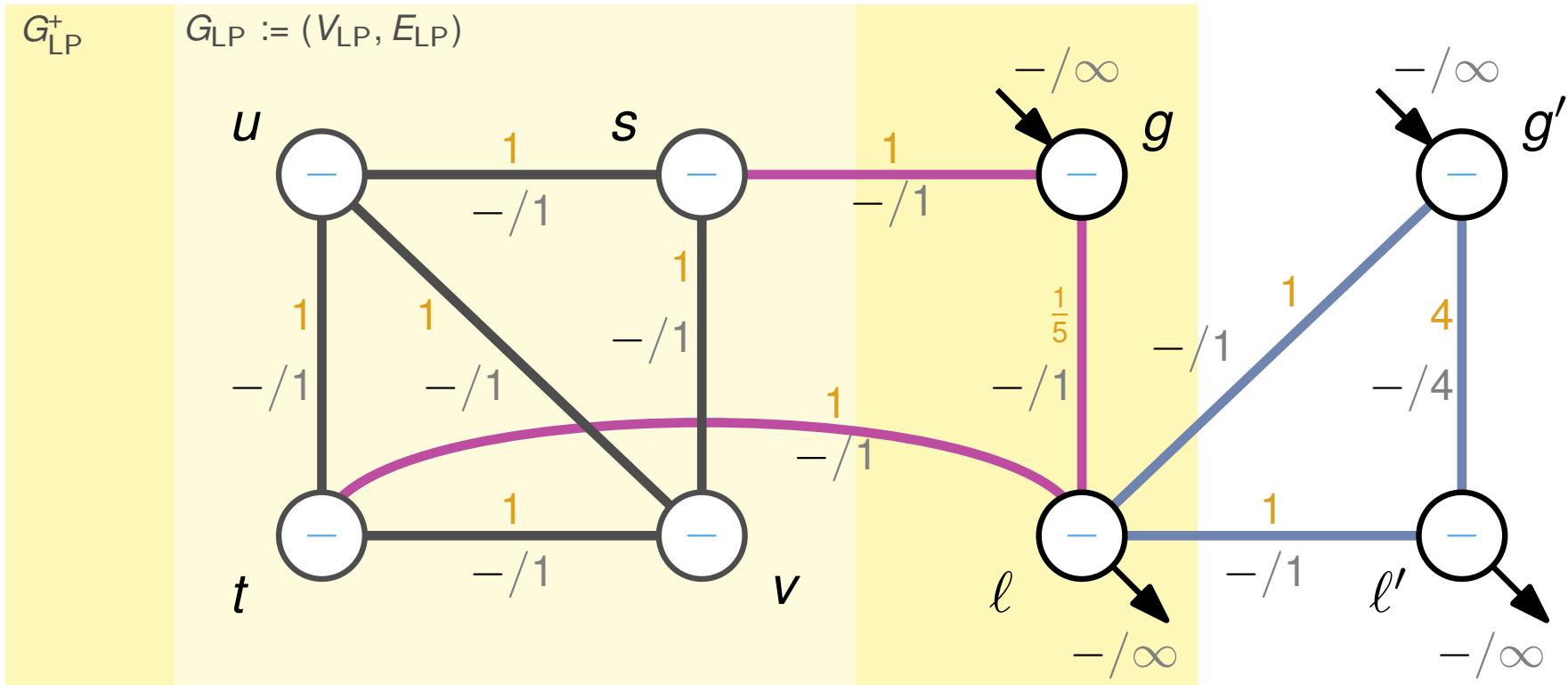
It is not possible to approximate the MTSF problem within a factor of $2^{(\log n)^{1-\epsilon}}$ unless NP is contained within quasi-polynomial deterministic time.

MTSF on Arbitrary Graphs is non-APX

[Lehmann et al., 2014]

Instance $\mathcal{N} = (G = (V \cup V_{LP}, E \cup E_{LP}), V_G = \{g, g'\}, V_D = \{\ell, \ell'\}, \text{cap}, b, \bar{x}, \underline{d}_\ell = 3, \bar{d})$

- Set of vertices $V_{LP} = \{v_1, v_2, \dots, v_n\}$, and $\bar{x} = \bar{d} = \infty$,
- Set of edges $E = E_{LP} \cup \{(g, s), (t, \ell), (g, \ell), (g', \ell'), (g', \ell), (\ell', \ell)\}$,
- Edge parameters $\text{cap}(e) := b(e) := 1 \quad \forall e \in E \setminus \{(g, \ell), (g', \ell')\}$,
 $\text{cap}(g, \ell) := 1, b(g, \ell) := \frac{1}{n+1},$
 $\text{cap}(g', \ell') := n, b(g', \ell') := n$

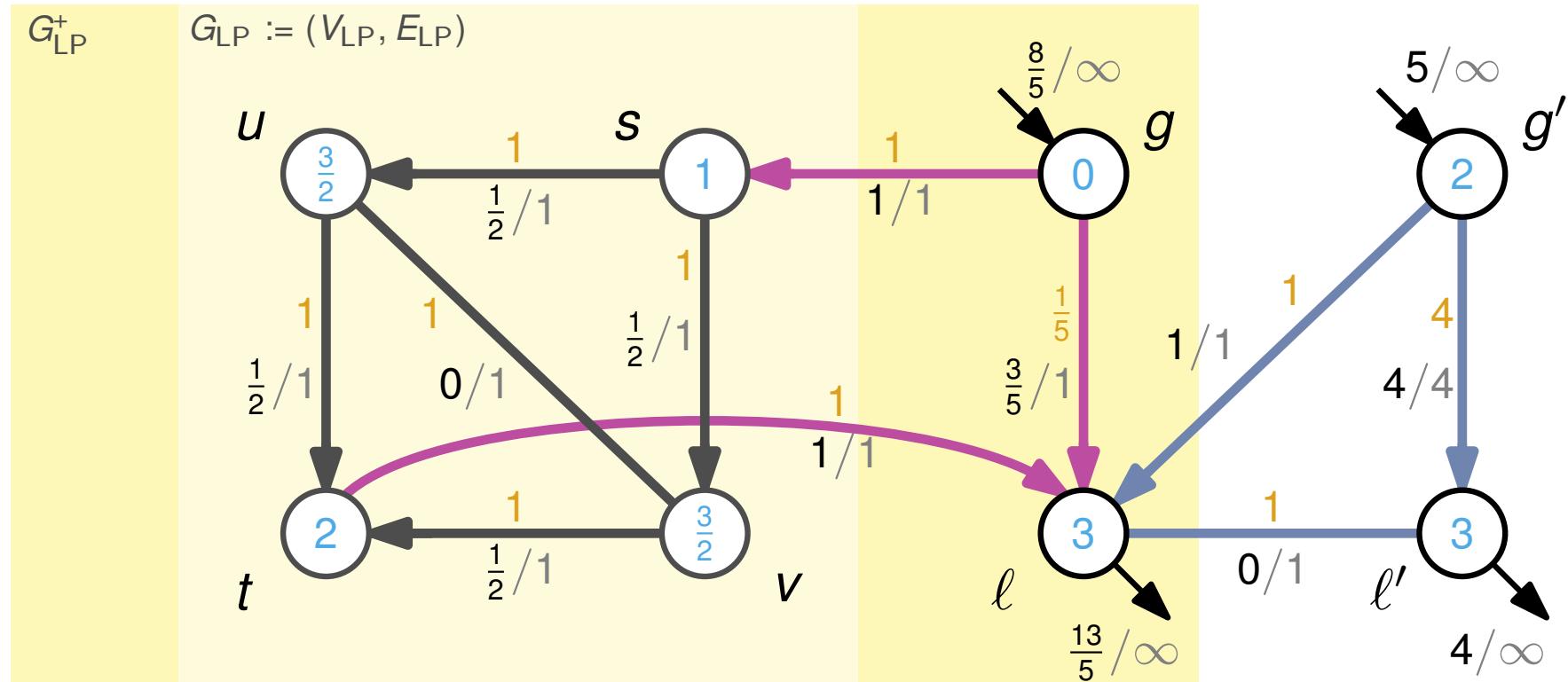


MTSF on Arbitrary Graphs is non-APX

[Lehmann et al., 2014]

Instance $\mathcal{N} = (G = (V \cup V_{LP}, E \cup E_{LP}), V_G = \{g, g'\}, V_D = \{\ell, \ell'\}, \text{cap}, \underline{b}, \bar{x}, \underline{d}_\ell = 3, \bar{d})$

- Set of vertices $V_{LP} = \{v_1, v_2, \dots, v_n\}$, and $\bar{x} = \bar{d} = \infty$,
- Set of edges $E = E_{LP} \cup \{(g, s), (t, \ell), (g, \ell), (g', \ell'), (g', \ell), (\ell', \ell)\}$,
- Edge parameters $\text{cap}(e) := b(e) := 1 \quad \forall e \in E \setminus \{(g, \ell), (g', \ell')\}$,
- $\text{cap}(g, \ell) := 1, b(g, \ell) := \frac{1}{n+1},$
- $\text{cap}(g', \ell') := n, b(g', \ell') := n$



$$\text{OPT}_{\text{MPF}}(\mathcal{N}) = \frac{33}{5}$$

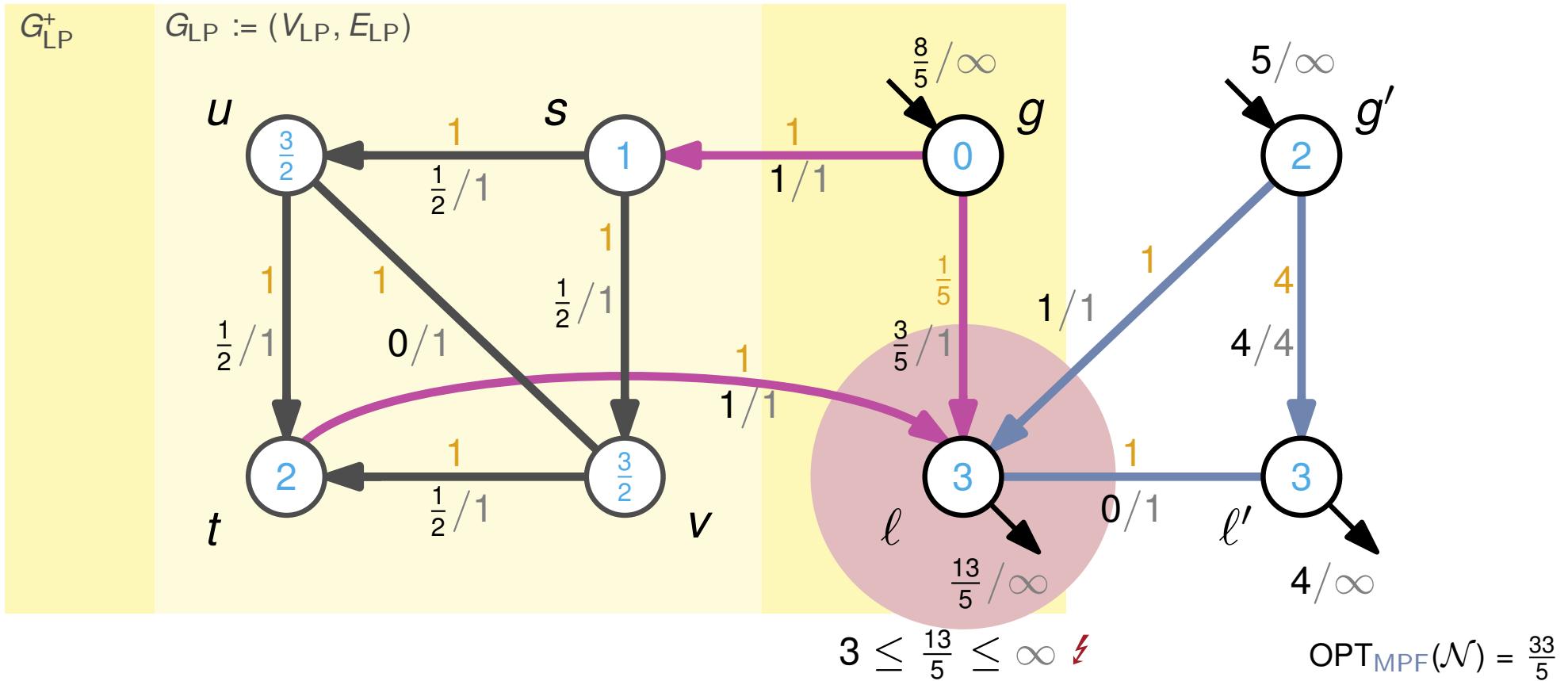
MTSF on Arbitrary Graphs is non-APX

[Lehmann et al., 2014]

Instance $\mathcal{N} = (G = (V \cup V_{LP}, E \cup E_{LP}), V_G = \{g, g'\}, V_D = \{\ell, \ell'\}, \text{cap}, b, \bar{x}, \underline{d}_\ell = 3, \bar{d})$

⚡

- Set of vertices $V_{LP} = \{v_1, v_2, \dots, v_n\}$, and $\bar{x} = \bar{d} = \infty$,
- Set of edges $E = E_{LP} \cup \{(g, s), (t, \ell), (g, \ell), (g', \ell'), (g', \ell), (\ell', \ell)\}$,
- Edge parameters $\text{cap}(e) := b(e) := 1 \quad \forall e \in E \setminus \{(g, \ell), (g', \ell')\}$,
- $\text{cap}(g, \ell) := 1, b(g, \ell) := \frac{1}{n+1},$
- $\text{cap}(g', \ell') := n, b(g', \ell') := n$

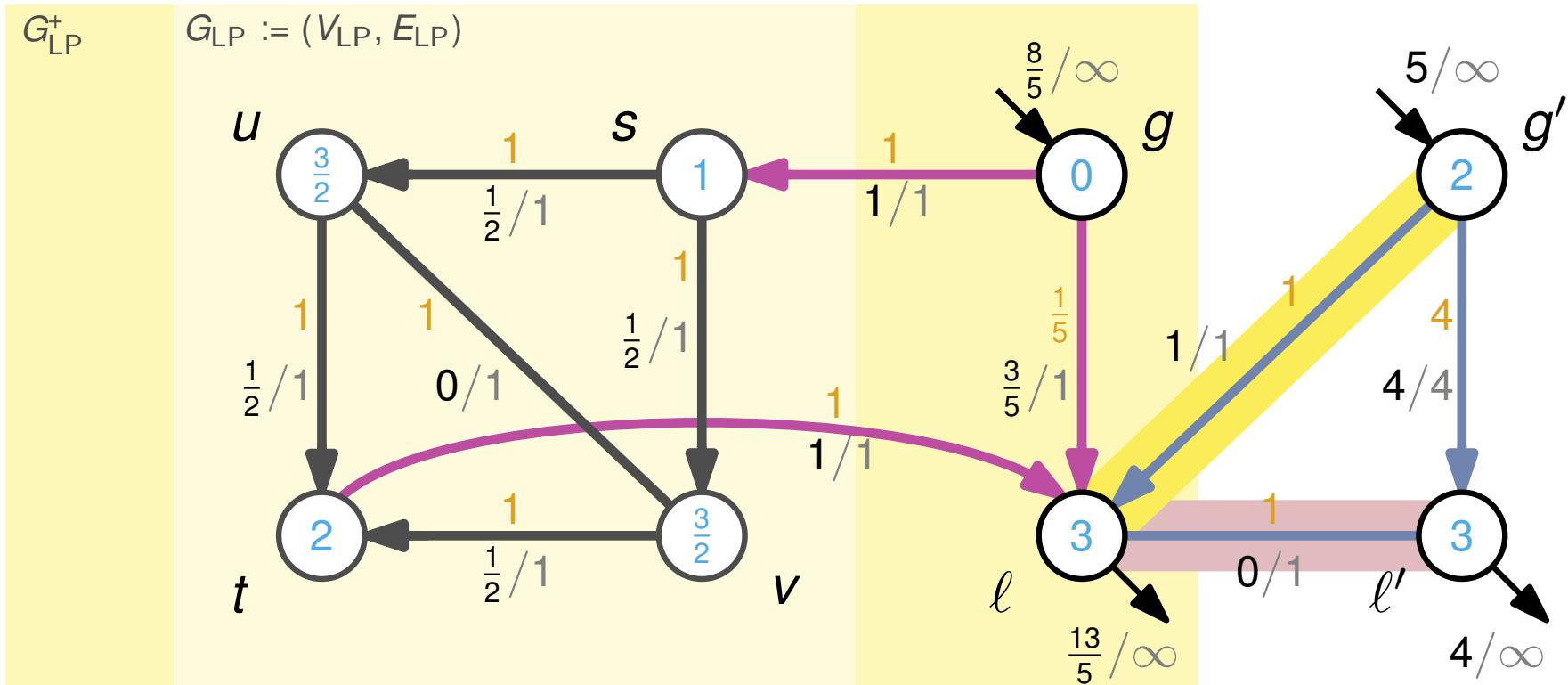


MTSF on Arbitrary Graphs is non-APX

[Lehmann et al., 2014]

Instance $\mathcal{N} = (G = (V \cup V_{LP}, E \cup E_{LP}), V_G = \{g, g'\}, V_D = \{\ell, \ell'\}, \text{cap}, \underline{b}, \bar{x}, \underline{d}_\ell = 3, \bar{d})$

- Set of vertices $V_{LP} = \{v_1, v_2, \dots, v_n\}$, and $\bar{x} = \bar{d} = \infty$,
- Set of edges $E = E_{LP} \cup \{(g, s), (t, \ell), (g, \ell), (g', \ell'), (g', \ell), (\ell', \ell)\}$,
- Edge parameters $\text{cap}(e) := b(e) := 1 \quad \forall e \in E \setminus \{(g, \ell), (g', \ell')\}$,
- $\text{cap}(g, \ell) := 1, b(g, \ell) := \frac{1}{n+1},$
- $\text{cap}(g', \ell') := n, b(g', \ell') := n$



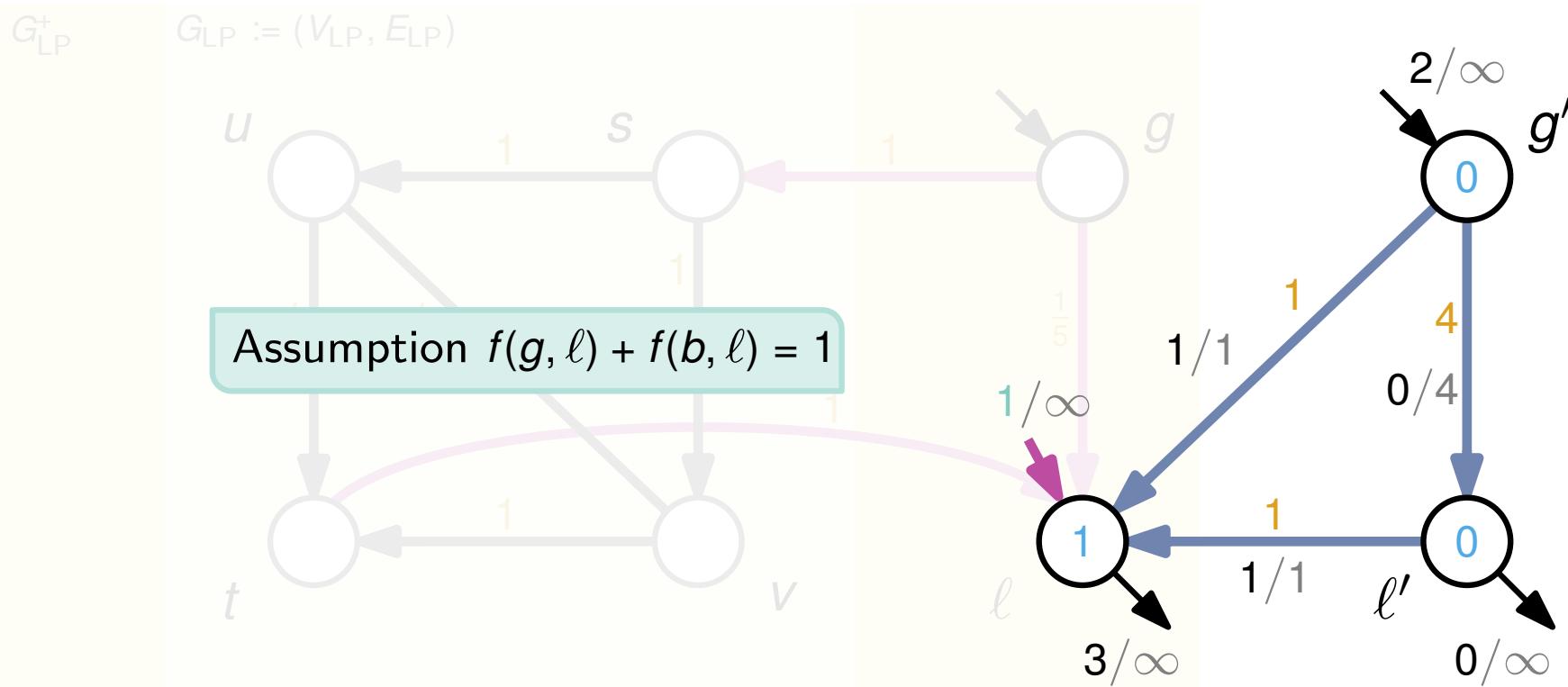
$$\text{OPT}_{\text{MPF}}(\mathcal{N}) = \frac{33}{5}$$

MTSF on Arbitrary Graphs is non-APX

[Lehmann et al., 2014]

Instance $\mathcal{N} = (G = (V \cup V_{LP}, E \cup E_{LP}), V_G = \{g, g'\}, V_D = \{\ell, \ell'\}, \text{cap}, b, \bar{x}, \underline{d}_\ell = 3, \bar{d})$

- Set of vertices $V_{LP} = \{v_1, v_2, \dots, v_n\}$, and $\bar{x} = \bar{d} = \infty$,
- Set of edges $E = E_{LP} \cup \{(g, s), (t, \ell), (g, \ell), (g', \ell'), (g', \ell), (\ell', \ell)\}$,
- Edge parameters $\text{cap}(e) := b(e) := 1 \quad \forall e \in E \setminus \{(g, \ell), (g', \ell')\}$,
 $\text{cap}(g, \ell) := 1, b(g, \ell) := \frac{1}{n+1},$
 $\text{cap}(g', \ell') := n, b(g', \ell') := n$

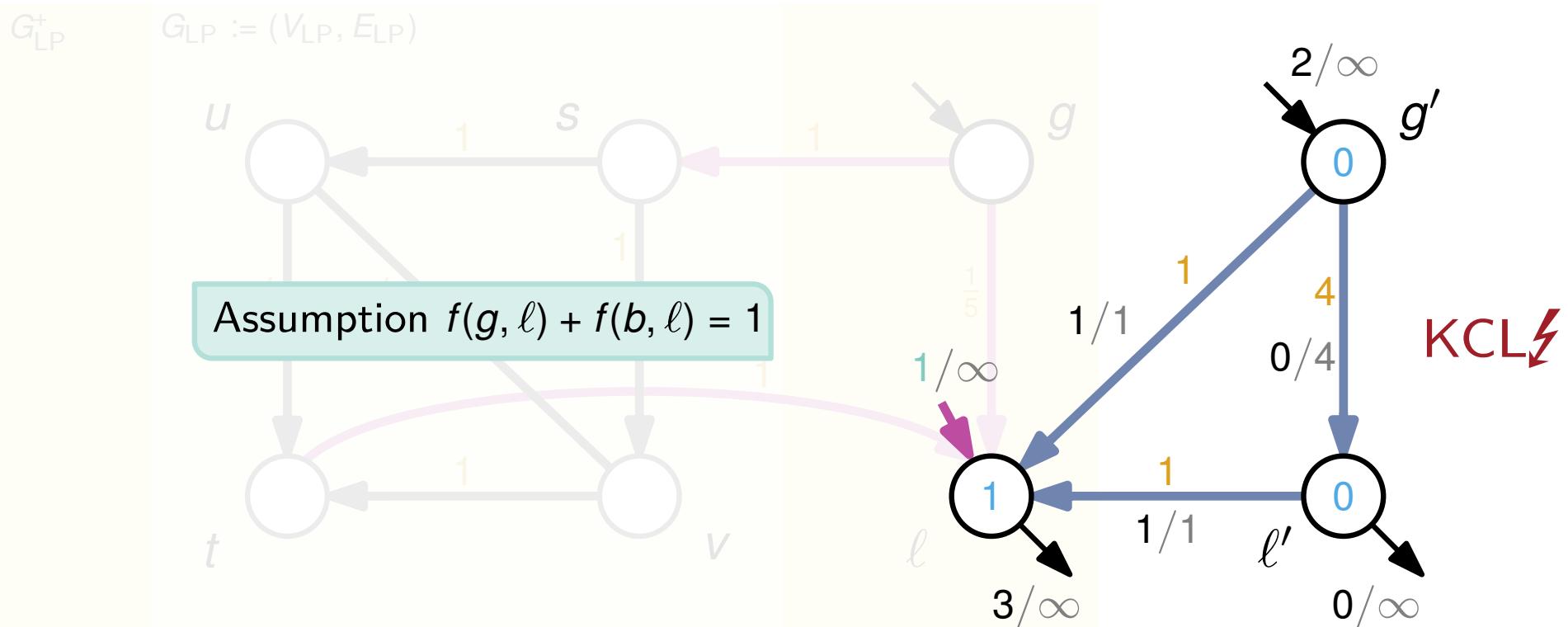


MTSF on Arbitrary Graphs is non-APX

[Lehmann et al., 2014]

Instance $\mathcal{N} = (G = (V \cup V_{LP}, E \cup E_{LP}), V_G = \{g, g'\}, V_D = \{\ell, \ell'\}, \text{cap}, b, \bar{x}, \underline{d}_\ell = 3, \bar{d})$

- Set of vertices $V_{LP} = \{v_1, v_2, \dots, v_n\}$, and $\bar{x} = \bar{d} = \infty$,
- Set of edges $E = E_{LP} \cup \{(g, s), (t, \ell), (g, \ell), (g', \ell'), (g', \ell), (\ell', \ell)\}$,
- Edge parameters $\text{cap}(e) := b(e) := 1 \quad \forall e \in E \setminus \{(g, \ell), (g', \ell')\}$,
 $\text{cap}(g, \ell) := 1, b(g, \ell) := \frac{1}{n+1},$
 $\text{cap}(g', \ell') := n, b(g', \ell') := n$



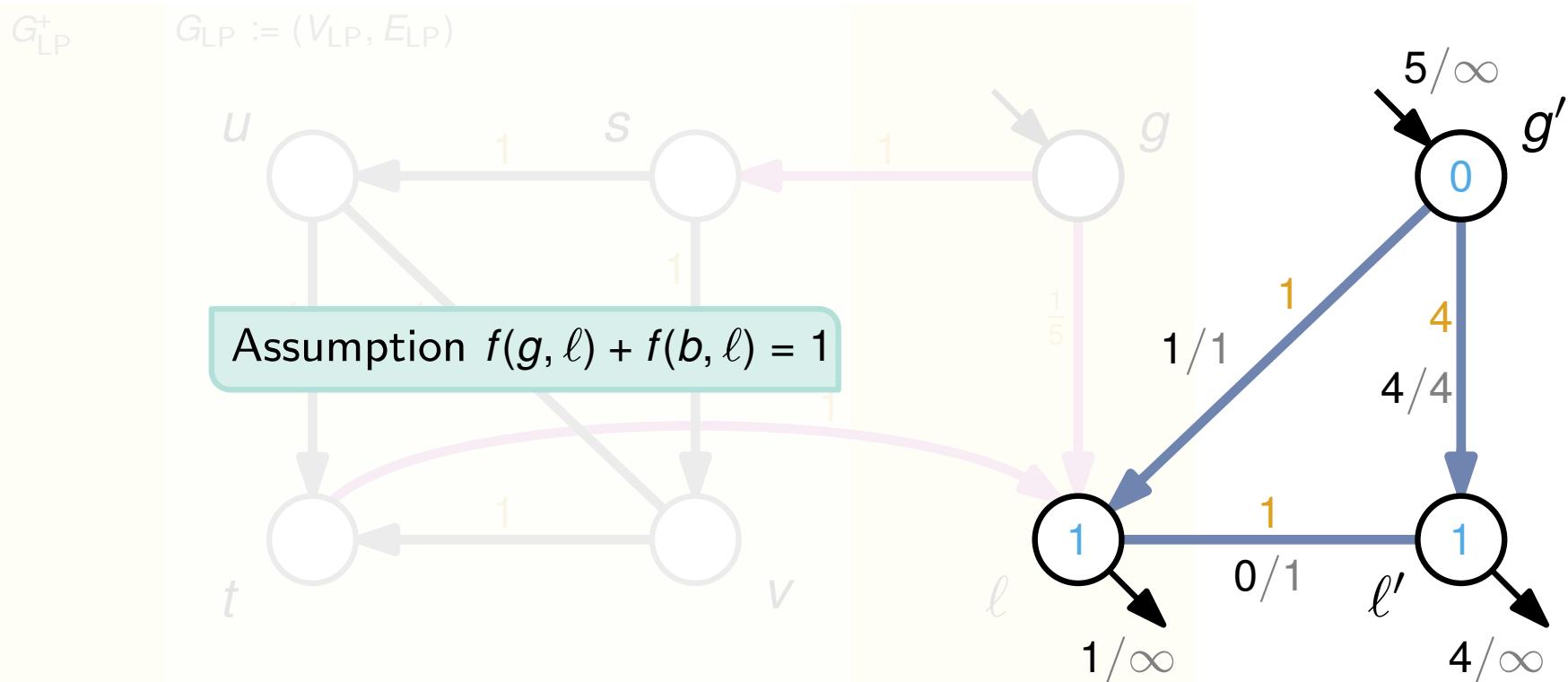
What is the maximum possible power flow in $(\mathcal{N} \setminus G_{LP}) - g$?

MTSF on Arbitrary Graphs is non-APX

[Lehmann et al., 2014]

Instance $\mathcal{N} = (G = (V \cup V_{LP}, E \cup E_{LP}), V_G = \{g, g'\}, V_D = \{\ell, \ell'\}, \text{cap}, b, \bar{x}, \underline{d}_\ell = 3, \bar{d})$

- Set of vertices $V_{LP} = \{v_1, v_2, \dots, v_n\}$, and $\bar{x} = \bar{d} = \infty$,
- Set of edges $E = E_{LP} \cup \{(g, s), (t, \ell), (g, \ell), (g', \ell'), (g', \ell), (\ell', \ell)\}$,
- Edge parameters $\text{cap}(e) := b(e) := 1 \quad \forall e \in E \setminus \{(g, \ell), (g', \ell')\}$,
 $\text{cap}(g, \ell) := 1, b(g, \ell) := \frac{1}{n+1},$
 $\text{cap}(g', \ell') := n, b(g', \ell') := n$



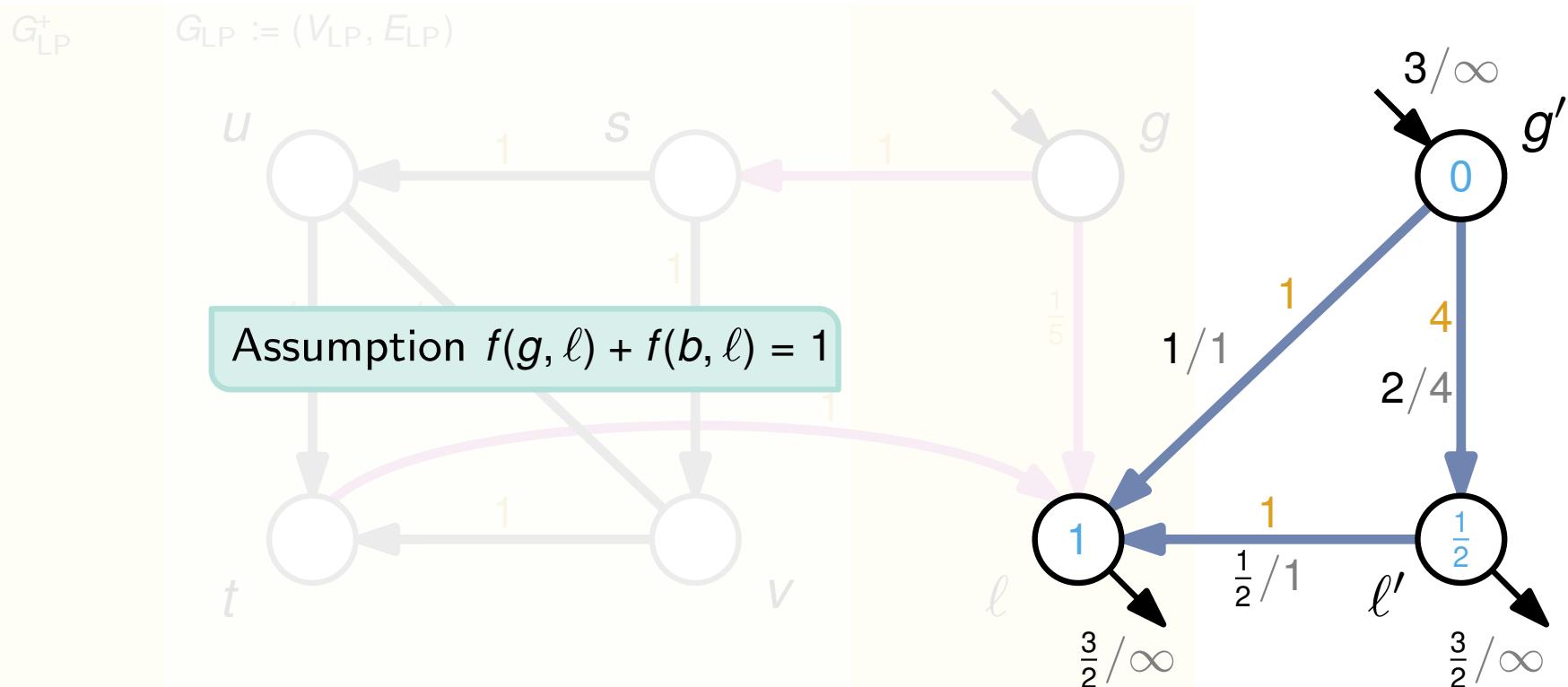
What is the maximum possible power flow in $(\mathcal{N} \setminus G_{LP}) - g$? $\text{OPT}_{\text{MPF}}((\mathcal{N} \setminus G_{LP}) - g) = 5$

MTSF on Arbitrary Graphs is non-APX

[Lehmann et al., 2014]

Instance $\mathcal{N} = (G = (V \cup V_{LP}, E \cup E_{LP}), V_G = \{g, g'\}, V_D = \{\ell, \ell'\}, \text{cap}, b, \bar{x}, \underline{d}_\ell = 3, \bar{d})$

- Set of vertices $V_{LP} = \{v_1, v_2, \dots, v_n\}$, and $\bar{x} = \bar{d} = \infty$,
- Set of edges $E = E_{LP} \cup \{(g, s), (t, \ell), (g, \ell), (g', \ell'), (g', \ell), (\ell', \ell)\}$,
- Edge parameters $\text{cap}(e) := b(e) := 1 \quad \forall e \in E \setminus \{(g, \ell), (g', \ell')\}$,
 $\text{cap}(g, \ell) := 1, b(g, \ell) := \frac{1}{n+1},$
 $\text{cap}(g', \ell') := n, b(g', \ell') := n$



What is the maximum possible demand for ℓ in $(\mathcal{N} \setminus G_{LP}) - g$?

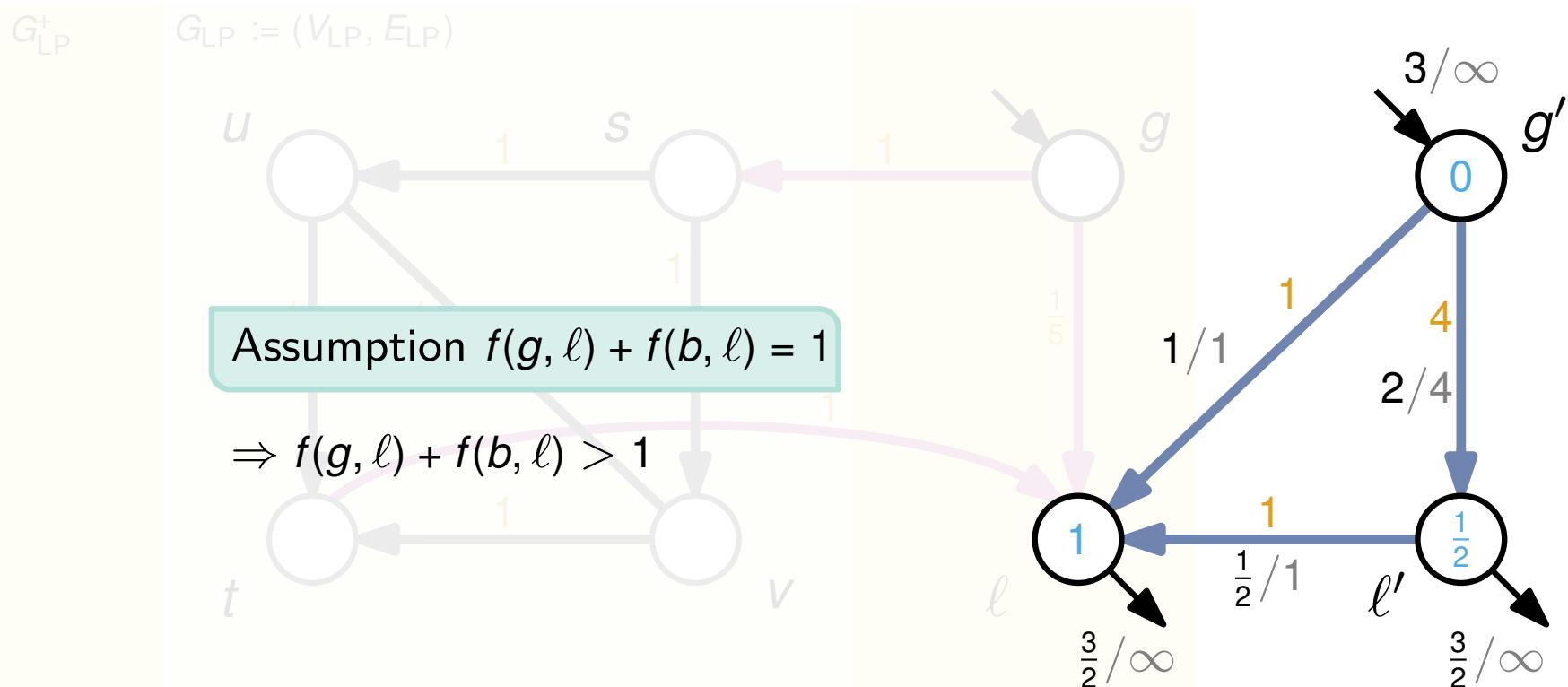
$$-\max f_{\text{net}}(\ell) = 3$$

MTSF on Arbitrary Graphs is non-APX

[Lehmann et al., 2014]

Instance $\mathcal{N} = (G = (V \cup V_{LP}, E \cup E_{LP}), V_G = \{g, g'\}, V_D = \{\ell, \ell'\}, \text{cap}, b, \bar{x}, \underline{d}_\ell = 3, \bar{d})$

- Set of vertices $V_{LP} = \{v_1, v_2, \dots, v_n\}$, and $\bar{x} = \bar{d} = \infty$,
- Set of edges $E = E_{LP} \cup \{(g, s), (t, \ell), (g, \ell), (g', \ell'), (g', \ell), (\ell', \ell)\}$,
- Edge parameters $\text{cap}(e) := b(e) := 1 \quad \forall e \in E \setminus \{(g, \ell), (g', \ell')\}$,
- $\text{cap}(g, \ell) := 1, b(g, \ell) := \frac{1}{n+1},$
- $\text{cap}(g', \ell') := n, b(g', \ell') := n$



What is the maximum possible demand for ℓ in $(\mathcal{N} \setminus G_{LP}) - g$?

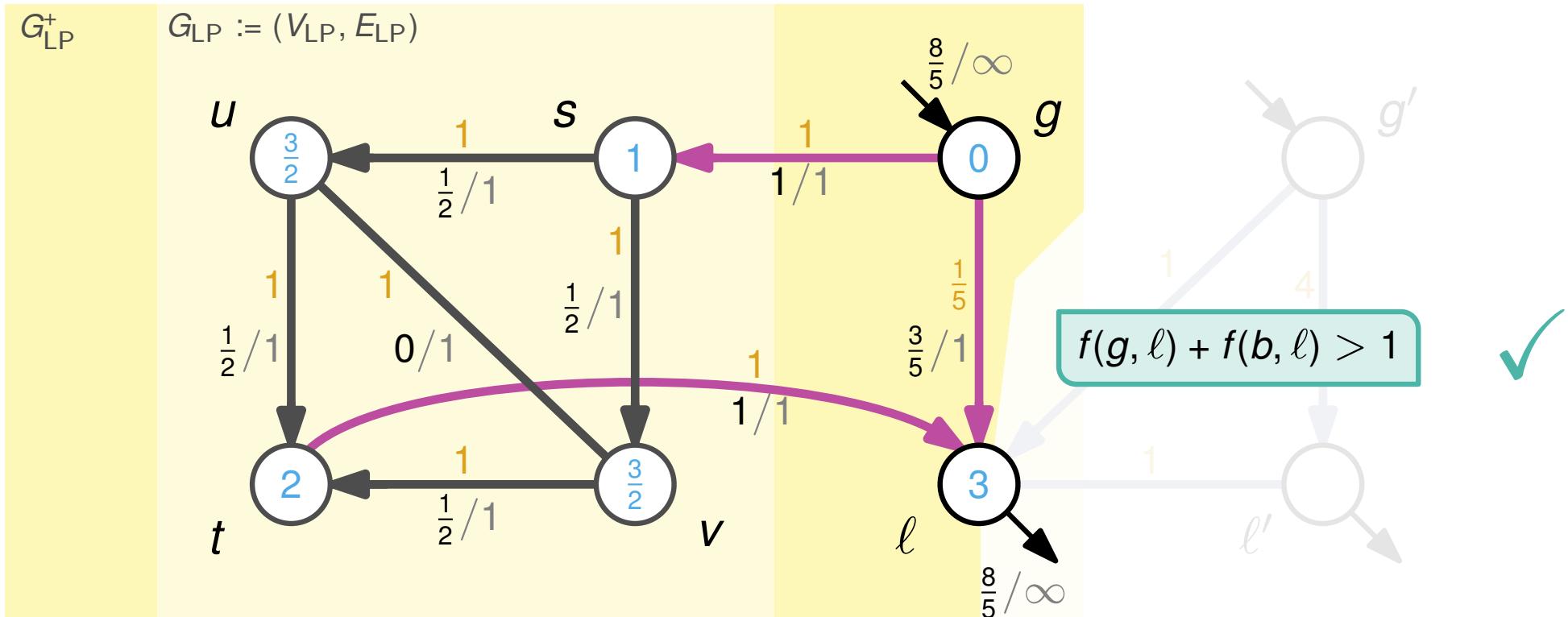
$- \max f_{\text{net}}(\ell) = 3$

MTSF on Arbitrary Graphs is non-APX

[Lehmann et al., 2014]

Instance $\mathcal{N} = (G = (V \cup V_{LP}, E \cup E_{LP}), V_G = \{g, g'\}, V_D = \{\ell, \ell'\}, \text{cap}, \mathbf{b}, \bar{x}, \underline{d}_\ell = 3, \bar{d})$

- Set of vertices $V_{LP} = \{v_1, v_2, \dots, v_n\}$, and $\bar{x} = \bar{d} = \infty$,
- Set of edges $E = E_{LP} \cup \{(g, s), (t, \ell), (g, \ell), (g', \ell'), (g', \ell), (\ell', \ell)\}$,
- Edge parameters $\text{cap}(e) := \mathbf{b}(e) := 1 \quad \forall e \in E \setminus \{(g, \ell), (g', \ell')\}$,
- $\text{cap}(g, \ell) := 1, \mathbf{b}(g, \ell) := \frac{1}{n+1},$
- $\text{cap}(g', \ell') := n, \mathbf{b}(g', \ell') := n$



G_{LP} limits the $\max \Delta\theta^\nu(s, t)$ that limits the maximum power flow in \mathcal{N}

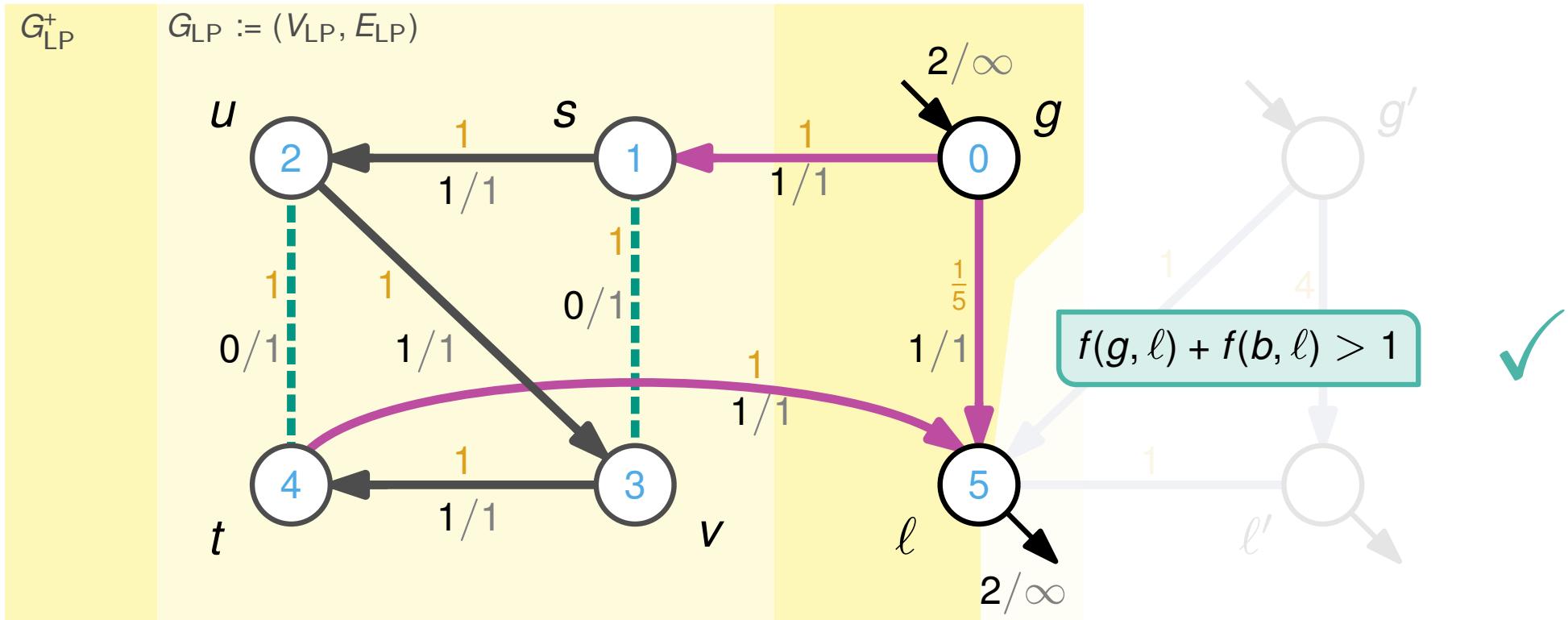
$$\text{OPT}_{\text{MPF}}(G_{LP}^+) = \frac{8}{5}$$

MTSF on Arbitrary Graphs is non-APX

[Lehmann et al., 2014]

Instance $\mathcal{N} = (G = (V \cup V_{LP}, E \cup E_{LP}), V_G = \{g, g'\}, V_D = \{\ell, \ell'\}, \text{cap}, \mathbf{b}, \bar{x}, \underline{d}_\ell = 3, \bar{d})$

- Set of vertices $V_{LP} = \{v_1, v_2, \dots, v_n\}$, and $\bar{x} = \bar{d} = \infty$,
- Set of edges $E = E_{LP} \cup \{(g, s), (t, \ell), (g, \ell), (g', \ell'), (g', \ell), (\ell', \ell)\}$,
- Edge parameters $\text{cap}(e) := \mathbf{b}(e) := 1 \quad \forall e \in E \setminus \{(g, \ell), (g', \ell')\}$,
- $\text{cap}(g, \ell) := 1, \mathbf{b}(g, \ell) := \frac{1}{n+1},$
- $\text{cap}(g', \ell') := n, \mathbf{b}(g', \ell') := n$

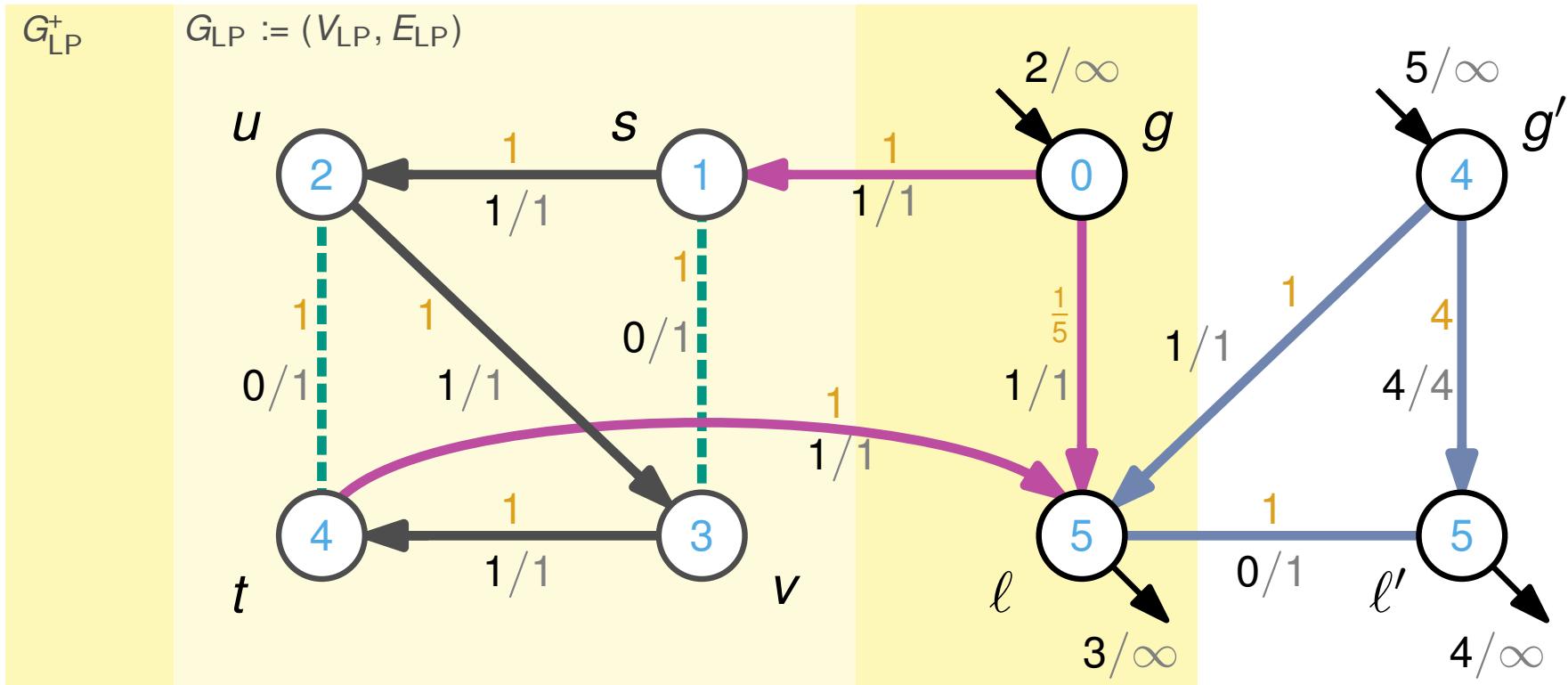


MTSF on Arbitrary Graphs is non-APX

[Lehmann et al., 2014]

Instance $\mathcal{N} = (G = (V \cup V_{LP}, E \cup E_{LP}), V_G = \{g, g'\}, V_D = \{\ell, \ell'\}, \text{cap}, \mathbf{b}, \bar{x}, \underline{d}_\ell = 3, \bar{d})$

- Set of vertices $V_{LP} = \{v_1, v_2, \dots, v_n\}$, and $\bar{x} = \bar{d} = \infty$,
- Set of edges $E = E_{LP} \cup \{(g, s), (t, \ell), (g, \ell), (g', \ell'), (g', \ell), (\ell', \ell)\}$,
- Edge parameters $\text{cap}(e) := \mathbf{b}(e) := 1 \quad \forall e \in E \setminus \{(g, \ell), (g', \ell')\}$,
- $\text{cap}(g, \ell) := 1, \mathbf{b}(g, \ell) := \frac{1}{n+1},$
- $\text{cap}(g', \ell') := n, \mathbf{b}(g', \ell') := n$



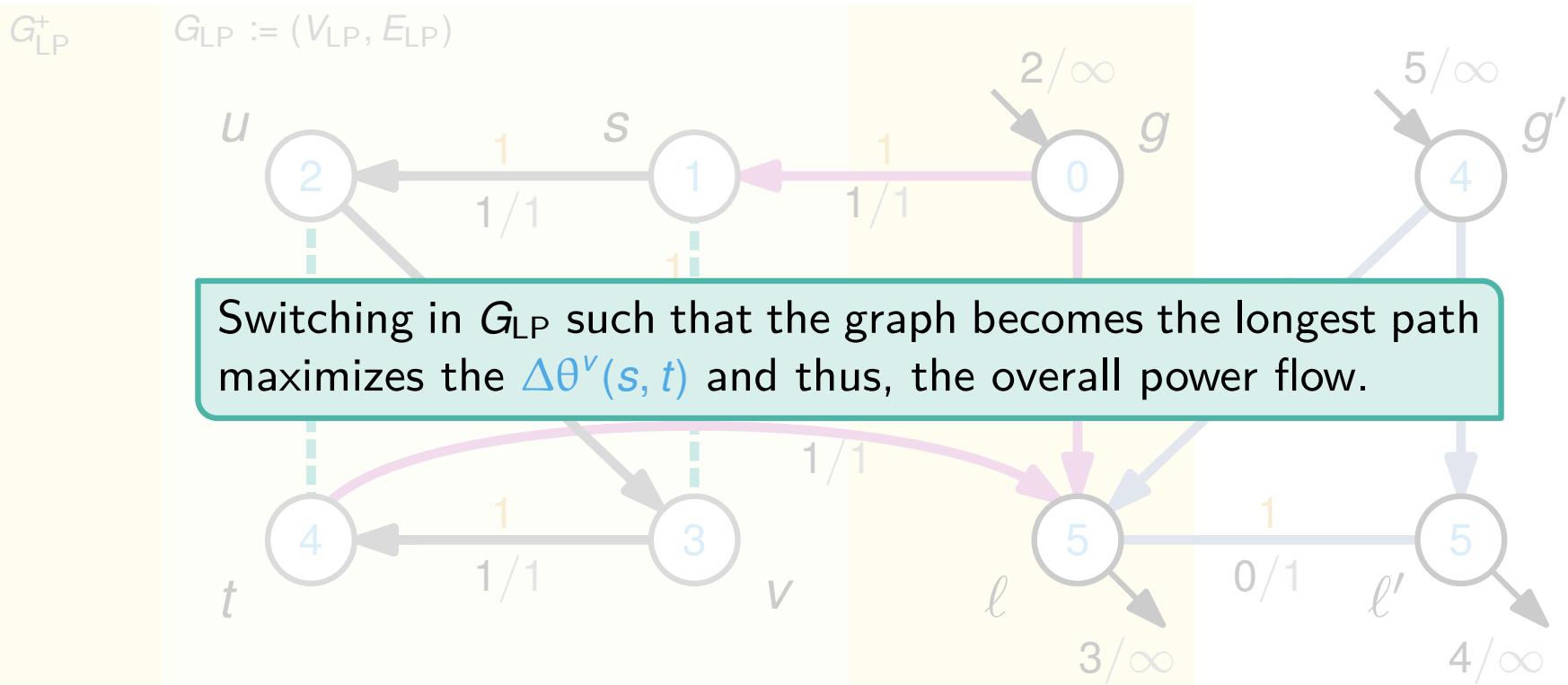
$$\text{OPT}_{\text{MTSF}}(\mathcal{N}) = 7$$

MTSF on Arbitrary Graphs is non-APX

[Lehmann et al., 2014]

Instance $\mathcal{N} = (G = (V \cup V_{LP}, E \cup E_{LP}), V_G = \{g, g'\}, V_D = \{\ell, \ell'\}, \text{cap}, b, \bar{x}, \underline{d}_\ell = 3, \bar{d})$

- Set of vertices $V_{LP} = \{v_1, v_2, \dots, v_n\}$, and $\bar{x} = \bar{d} = \infty$,
- Set of edges $E = E_{LP} \cup \{(g, s), (t, \ell), (g, \ell), (g', \ell'), (g', \ell), (\ell', \ell)\}$,
- Edge parameters $\text{cap}(e) := b(e) := 1 \quad \forall e \in E \setminus \{(g, \ell), (g', \ell')\}$,
 $\text{cap}(g, \ell) := 1, b(g, \ell) := \frac{1}{n+1},$
 $\text{cap}(g', \ell') := n, b(g', \ell') := n$



$$\text{OPT}_{\text{MTSF}}(\mathcal{N}) = 7$$

Summary & Future Work

	Graph Structure	Complexity	Algorithm
complexity ↓	penrose-minor-free graphs	polynomial-time solvable	✓
	series-parallel graphs	NP-hard	✗
	cacti with max degree of 3	NP-hard <small>[Lehmann et al. 2014]</small>	✓
	2-level trees	NP-hard <small>[Lehmann et al. 2014]</small>	✗
	planar graphs with max degree of 3	strongly NP-hard <small>[Lehmann et al. 2014]</small>	✗
	arbitrary graphs	non-APX <small>[Lehmann et al. 2014]</small>	✗

Legend:
 one generator, one load
 arbitrary generators, arbitrary loads
 $|V_G|=2, |V_D|=2$

Summary & Future Work

	Graph Structure	Complexity	Algorithm
one generator, one load	<p>penrose-minor-free graphs</p> <p>series-parallel graphs</p>  	<p>polynomial- time solvable</p> <p>NP-hard</p>	 
arbitrary $ V_G = 2, V_D = 2$	<ul style="list-style-type: none"> ■ What happens if we minimize the number of switches or fix a set of non-switchable edges? ■ Is there a PTAS on cacti for MTSF? ■ Replace X by ✓ 		
\geq -level trees		NP-hard [Lehmann et al. 2014]	
planar graphs with max degree of 3		strongly NP-hard [Lehmann et al. 2014]	
arbitrary graphs		non-APX [Lehmann et al. 2014]	

Results

NESTA Case	n	m	$ S_{MTSF} $	$ S_{MaxST} $	OPT_{PF}	OPT_{MPF}	OPT_{MTSF}	OPT_{MaxST}	OPT_{MF}	max Gen
3_1mbd	3	3	1	1	315.00	353.53	4 000.00	4 000.00	4 000.00	4 000.00
6_ww	6	11	6	6	210.00	332.80	360.00	360.00	470.00	530.00
9_wscc	9	9	6	2	315.00	770.00	770.00	770.00	770.00	820.00
24_ieee_rts	24	38	28	18	2 850.00	3 405.00	3 405.00	3 405.00	3 405.00	3 405.00
29_edin	29	99	55	79	56 325.90	81 597.50	81 603.40	76 158.80	82 384.80	82 384.80
30_as	30	41	32	15	283.40	435.00	435.00	435.00	435.00	435.00
39_epri	39	46	35	17	6 254.23	7 227.00	7 227.00	7 227.00	7 227.00	7 367.00
57_ieee	57	80	75	40	1 250.80	1 377.00	1 377.00	1 377.00	1 377.00	1 377.00
73_ieee_rts	73	120	87	56	8 550.00	10 215.00	10 215.00	10 215.00	10 215.00	10 215.00
189_edin	189	206	71	62	1 367.83	2 987.00	2 987.00	2 987.00	2 987.00	3 012.00
300_ieee	300	411	290	185	23 527.20	31 568.00	31 568.00	30 504.00	31 735.00	32 492.00
2736sp_mp	2736	3269	2518	1307	18 074.50	20 246.70	20 246.70	20 010.70	20 246.70	20 246.70
2737sop_mp	2737	3269	2536	1305	11 267.20	14 677.90	14 677.90	14 537.20	14 677.90	14 677.90
3120sp_mp	3120	3693	2793	1513	21 181.50	25 406.00	25 406.00	24 856.50	25 406.00	25 406.00

Open Problems

Problem	Graph Structure	$ V_G $	$ V_D $	b	cap	Complexity	Algorithms
MTSF and OTS	penrose-minor-free graphs	1	1	—	—	polynomial-time solvable [Grastien et al., 2018]	✓ better? $\infty \infty$
MTSF and OTS	series-parallel graphs	1	1	∞	∞	NP-hard [Kocuk et al., 2016]	✗
MTSF and OTS	cacti with max degree of 3	∞	∞	1	∞	NP-hard [Lehmann et al., 2014]	✓ better? $\infty \infty$
MTSF and OTS	2-level trees	1	∞	∞	∞	NP-hard [Lehmann et al., 2014]	✗
MTSF and OTS	planar graphs with max degree of 3	1	1	∞	1	strongly NP-hard [Lehmann et al., 2014]	✗
MTSF OTS	arbitrary graphs	2	2	∞	∞	non-APX [Lehmann et al., 2014]	✗
Other interesting structures?							

Any algorithm giving guarantees?

References

1. *Power systems test case archive*. University of Washington, Departement of Electrical Engineering, 1999. <https://labs.ece.uw.edu/pstca/>, Accessed: 2017-11-14.
2. Ray D. Zimmerman, Carlos E. Murillo-Sanchez, and Robert J. Thomas. *Matpower: Steady-state operations, planning, and analysis tools for power systems research and education*. IEEE Transactions on Power Systems, 26(1):12–19. DOI: 10.1109/TPWRS.2010.2051168, 2011.
3. Emily B. Fisher, Richard P. O'Neill, and Michael C. Ferris. *Optimal transmission switching*. IEEE Transactions on Power Systems, 23(3):1346–1355, 2008. DOI: 10.1109/TPWRS.2008.922256.
4. Michael R. Garey and David S. Johnson. *Computers and Intractability; a Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA. ISBN 0-716-71045-5, 1990.
5. Alban Grastien, Ignaz Rutter, Dorothea Wagner, Franziska Wegner, and Matthias Wolf. *The Maximum Transmission Switching Flow Problem*. In Proceedings of the Ninth International Conference on Future Energy Systems (e-Energy). ACM, New York, NY, USA, 340–360. DOI: 10.1145/3208903.3208910, 2018.
6. David R. Karger, Rajeev Motwani, and Gurumurthy D. S. Ramkumar. *On Approximating the Longest Path in a Graph*. Algorithmica. 18:1, 82–98, DOI: 10.1007/BF02523689, 1997.
7. Richard M. Karp. *Reducibility Among Combinatorial Problems*. In Complexity of Computer Computations, Proc. Sympos. IBM Thomas J. Watson Res. Center, Yorktown Heights, N.Y. (Ed. R. E. Miller and J. W. Thatcher). New York: Plenum, pp. 85–103. ISBN 0-306-30707-3, 1972.
8. Burak Kocuk, Hyemin Jeon, Santanu S. Dey, Jeff Linderoth, James Luedtke, and Xu Andy Sun. *A Cycle-Based Formulation and Valid Inequalities for DC Power Transmission Problems with Switching*. Operations Research. 64:4, 922–938, DOI: 10.1287/opre.2015.1471, 2016.

References

9. Karsten Lehmann, Alban Grastien, and Pascal Van Hentenryck. *The Complexity of DC-switching Problems*. CoRR, abs/1411.4369, 2014.
10. Karsten Lehmann, Alban Grastien, and Pascal Van Hentenryck. *The Complexity of Switching and FACTS Maximum-potential-flow Problems*. CoRR, abs/1507.04820, 2015.
11. Steven Skiena. *Implementing Discrete Mathematics: Combinatorics and Graph Theory with Mathematica*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA. ISBN 0-201-50943-1, 1991.
12. Kei Uchizawa, Takanori Aoki, Takehiro Ito, Akira Suzuki, and Xiao Zhou. *On the Rainbow Connectivity of Graphs: Complexity and FPT Algorithms*. Algorithmica, 67(2):161–179, DOI: 10.1007/s00453-012-9689-4, 2013.