Karlsruhe Institute of Technology

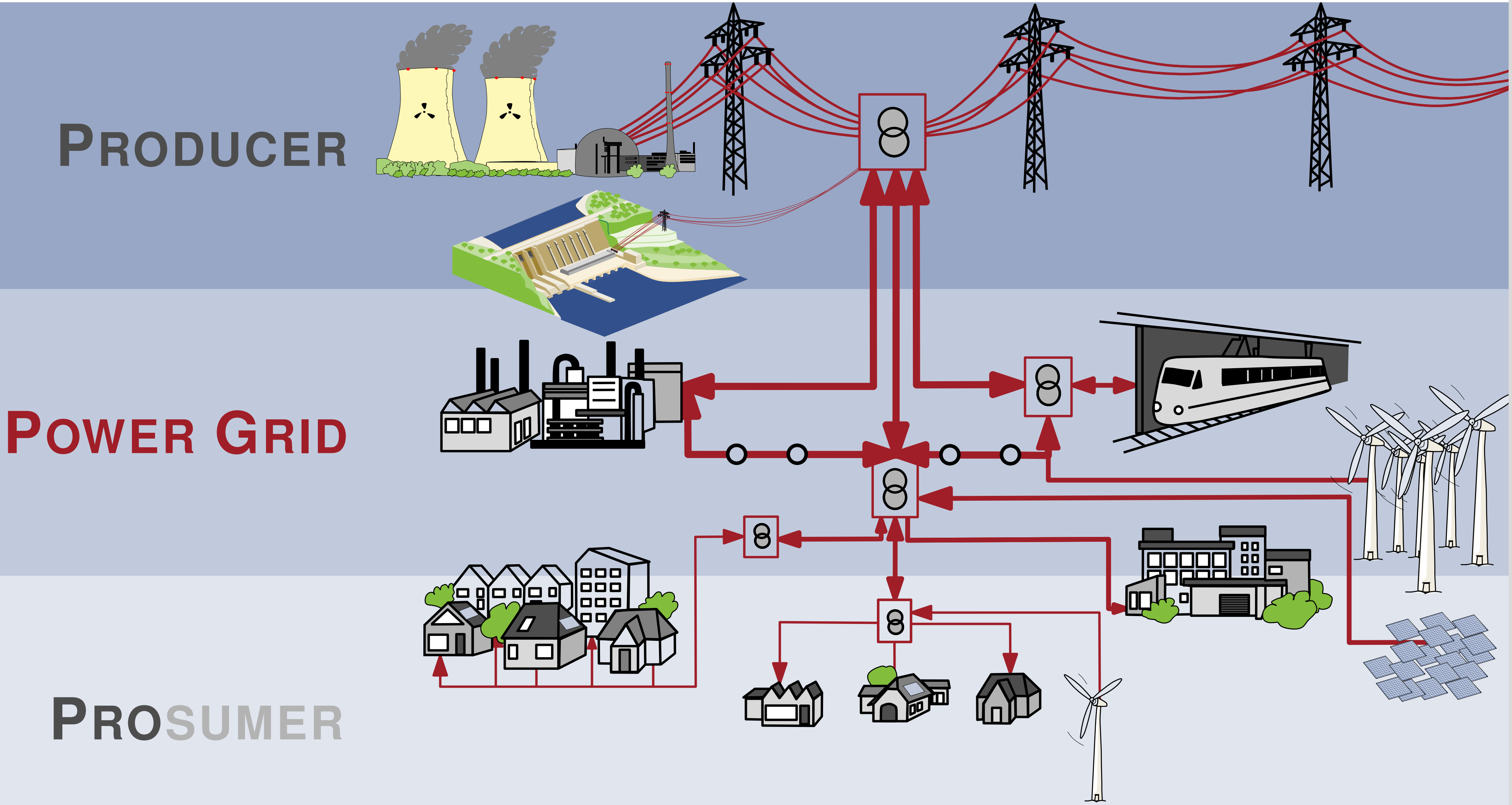
The Maximum Transmission Switching Flow Problem

Energy Informatics · Part 13 (L2) · July 18th, 2018

Franziska Wegner

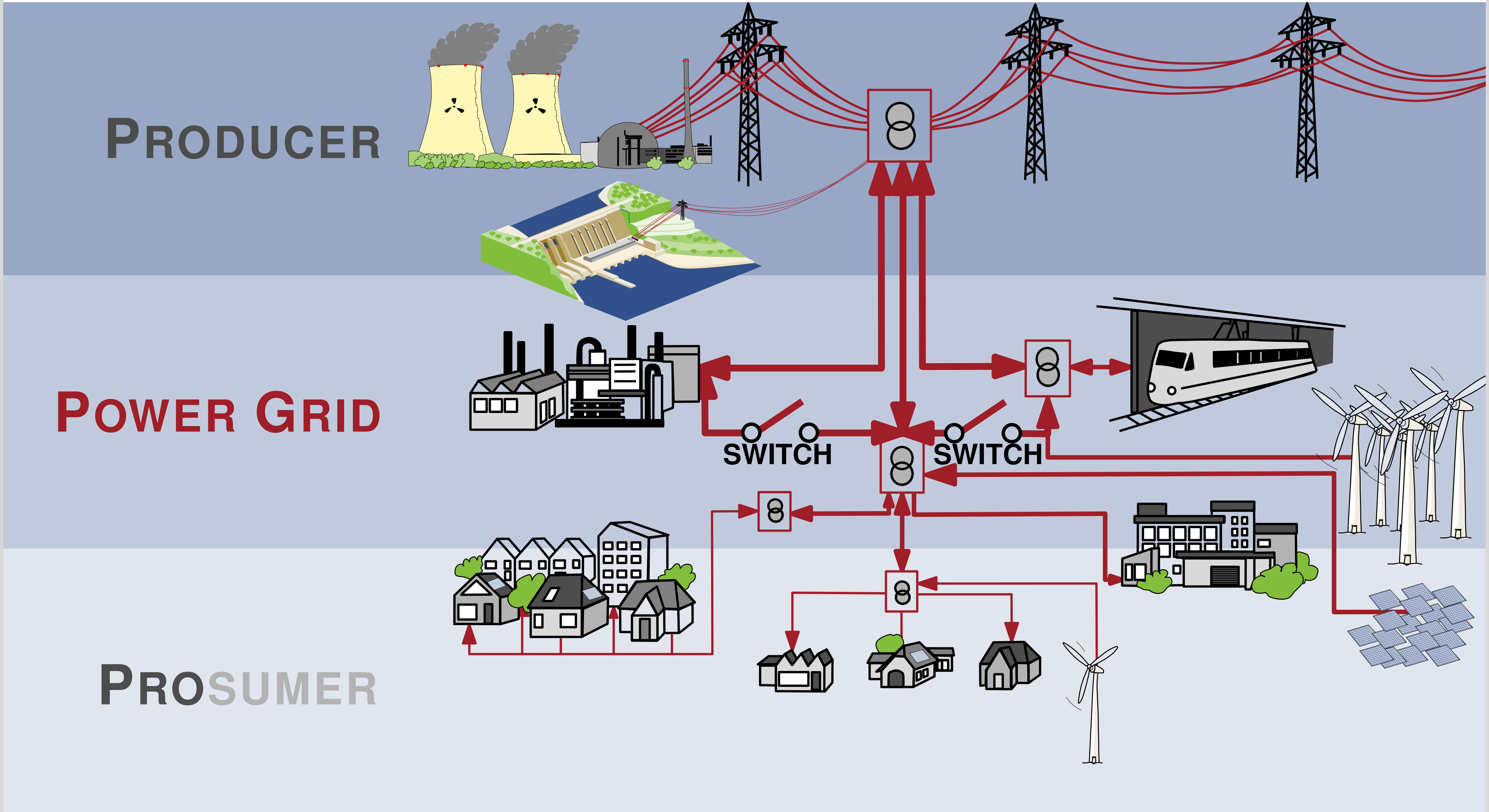
INSTITUTE OF THEORETICAL INFORMATICS · ALGORITHMICS GROUP

Recent Development in Power Grids



Institute of Theoretical Informatics
Algorithmics Group

Recent Development in Power Grids



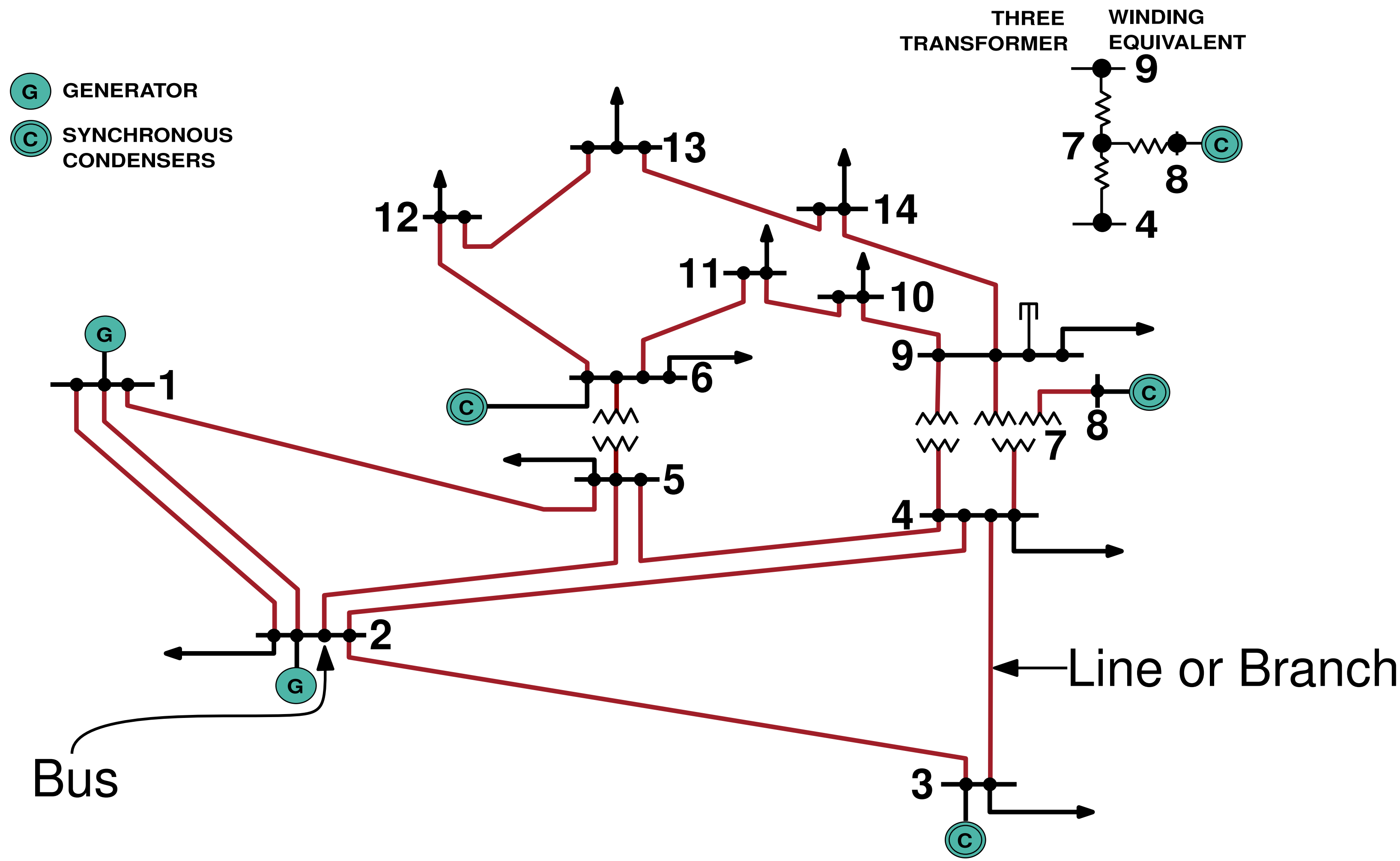
Institute of Theoretical Informatics
Algorithmics Group

From a Transmission Network to a Graph

[University of Washington, 1999]

Karlsruhe Institute of Technology

Graph $G = (V, E)$

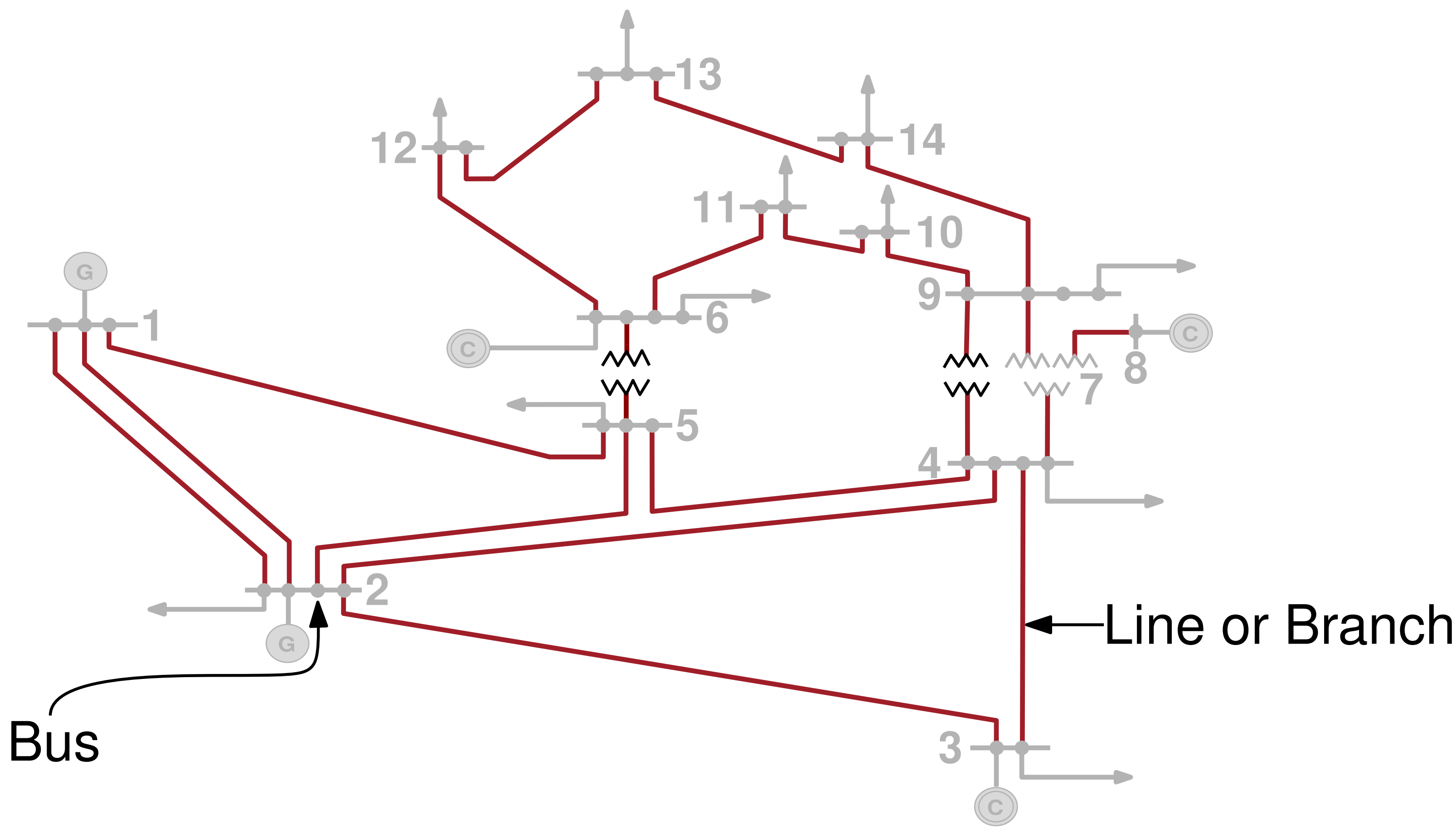


AEP 14 BUS TEST SYSTEM BUS CODE DIAGRAM

From a Transmission Network to a Graph

[University of Washington, 1999]

Graph $G = (V, E)$



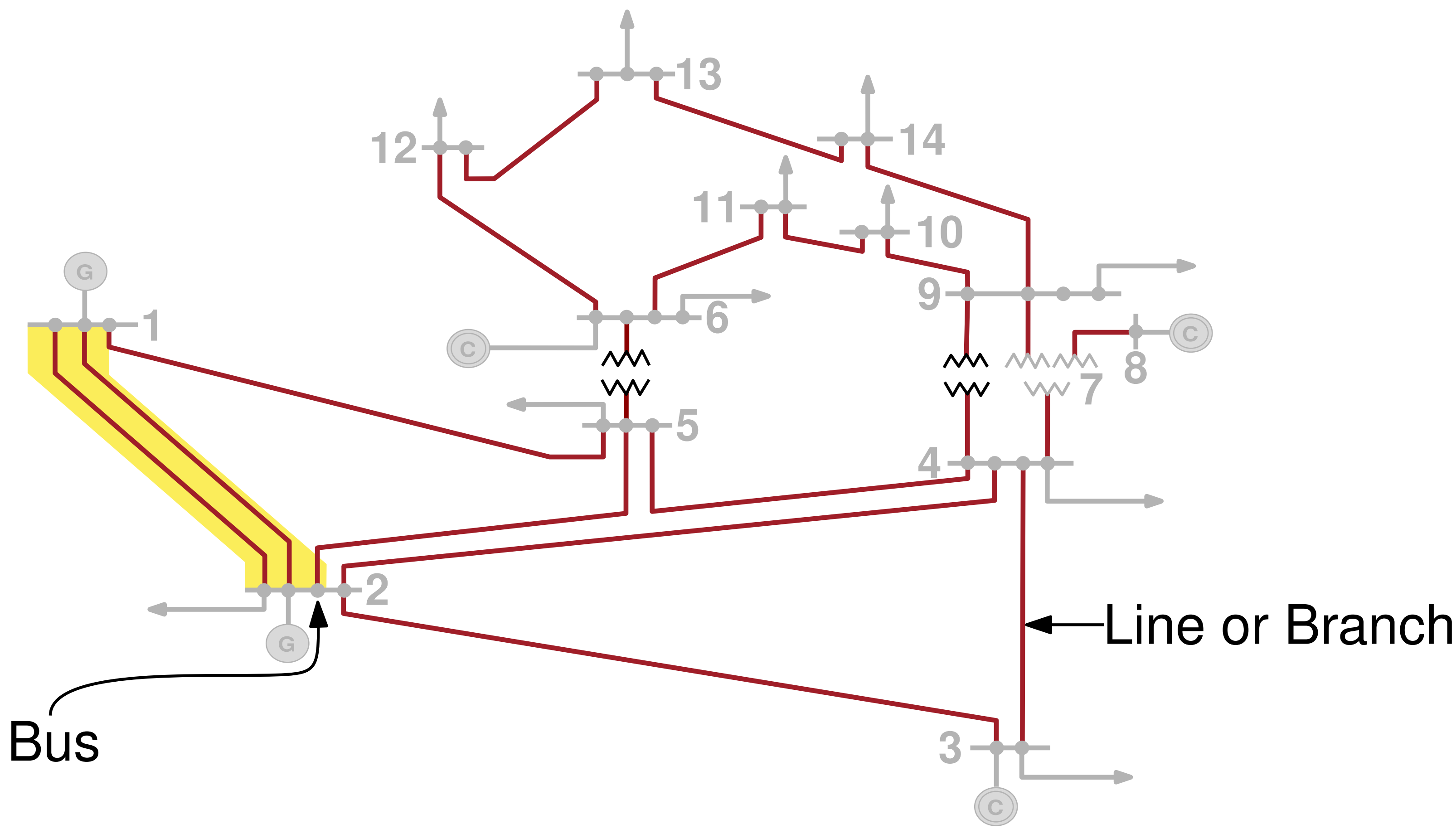
AEP 14 BUS TEST SYSTEM BUS CODE DIAGRAM

Institute of Theoretical Informatics
Algorithmics Group

From a Transmission Network to a Graph

[University of Washington, 1999]

Graph $G = (V, E)$

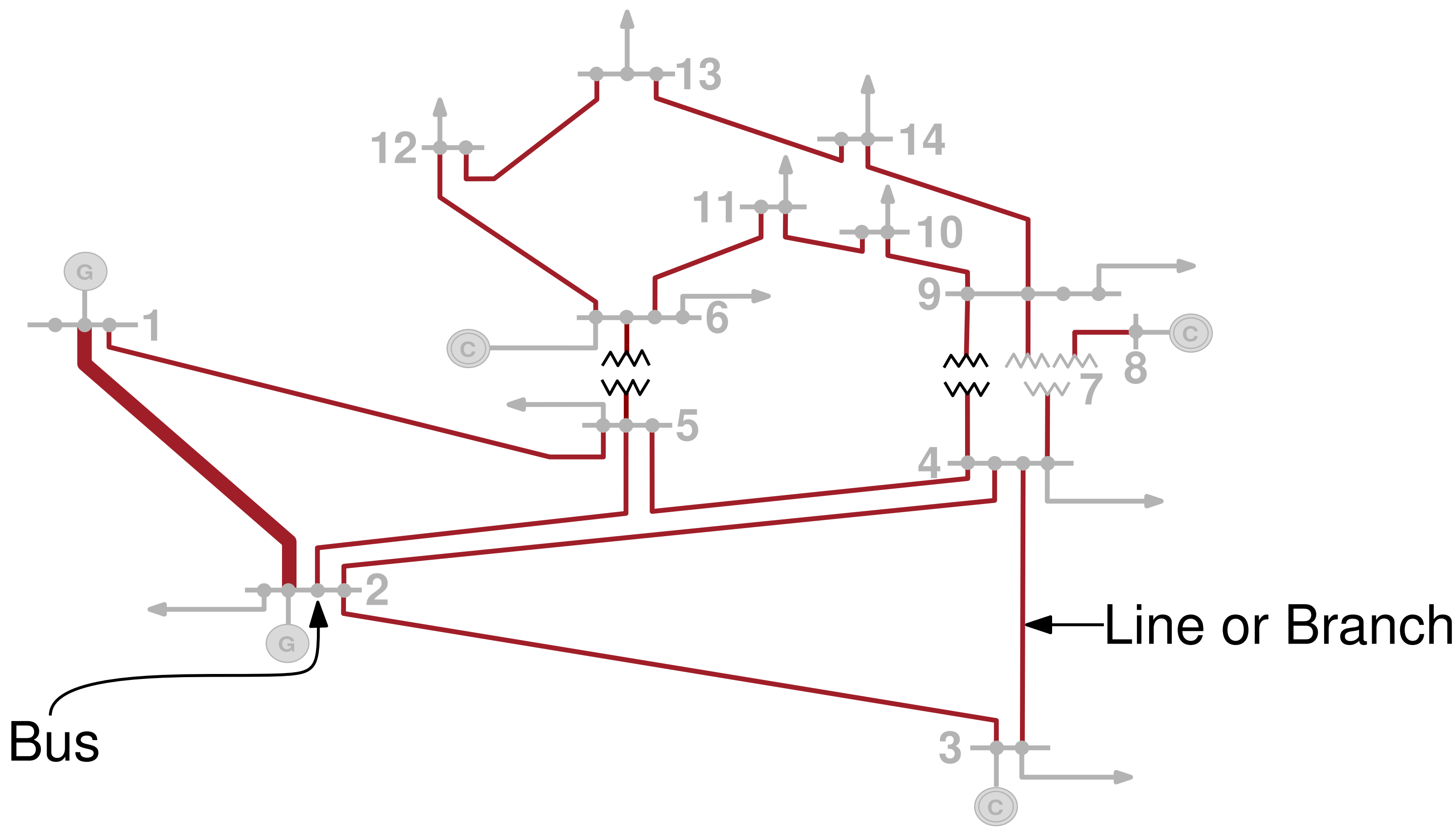


AEP 14 BUS TEST SYSTEM BUS CODE DIAGRAM

From a Transmission Network to a Graph

[University of Washington, 1999]

Graph $G = (V, E)$

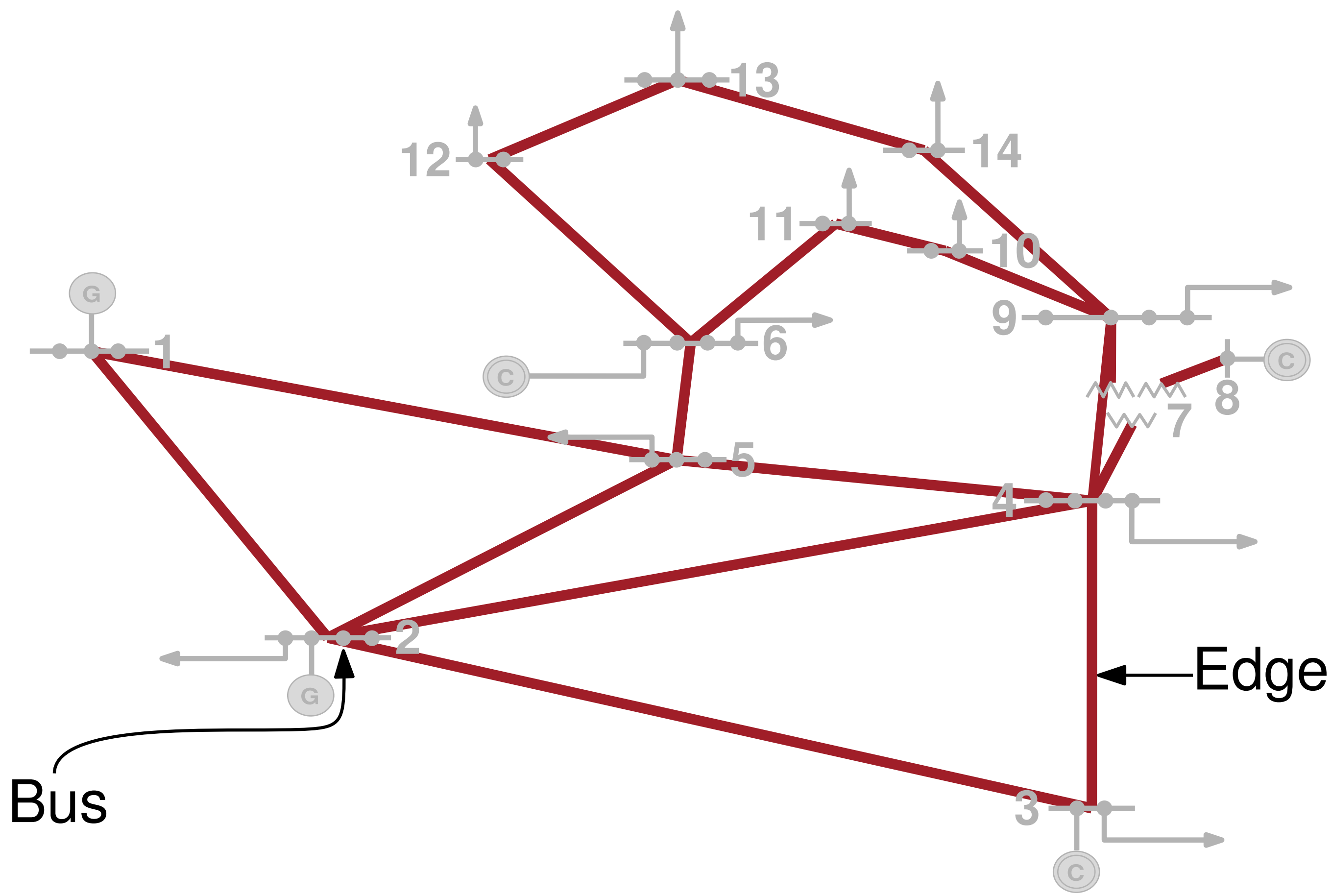


AEP 14 BUS TEST SYSTEM BUS CODE DIAGRAM

From a Transmission Network to a Graph

[University of Washington, 1999]

Graph G = (*V*, *E*)

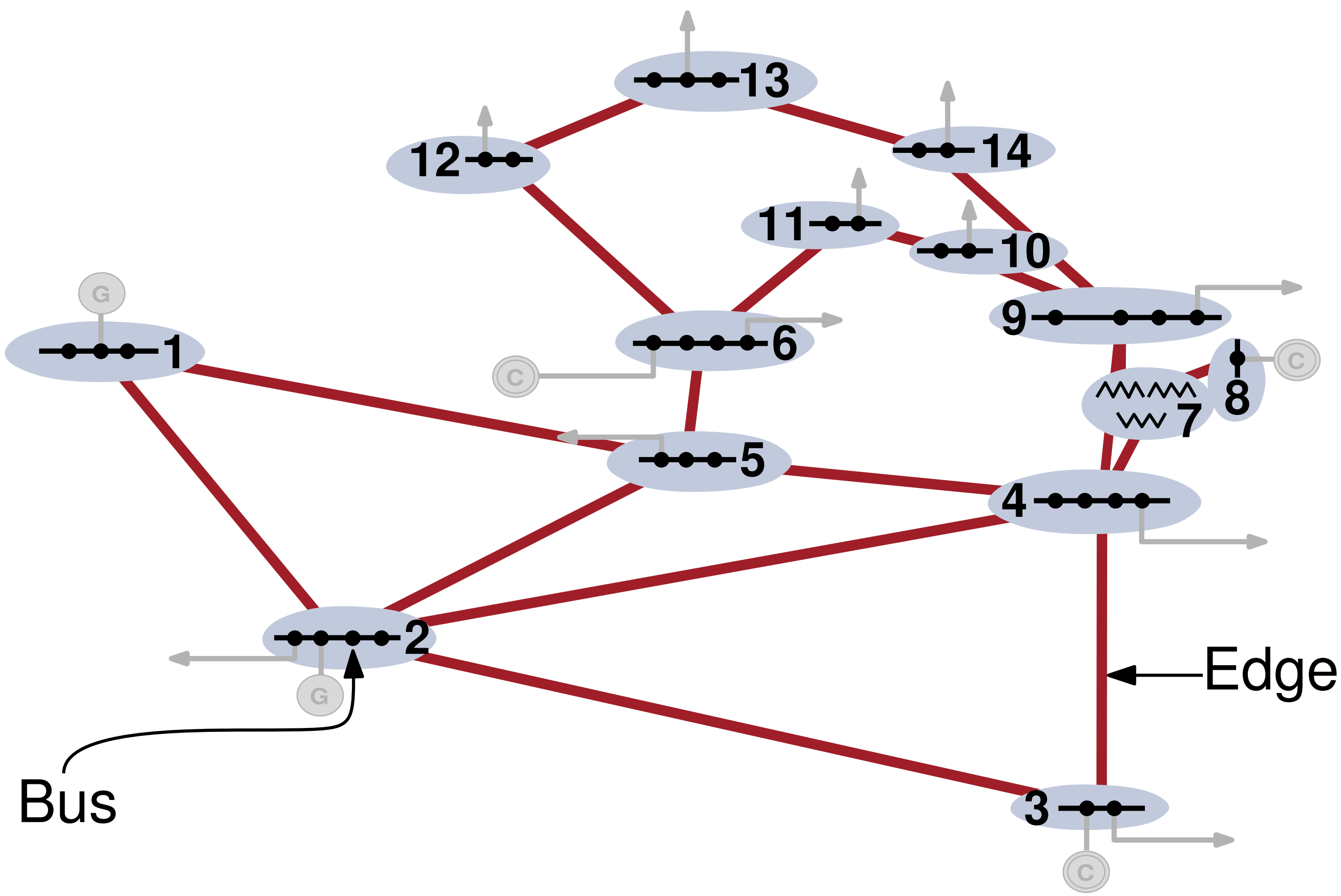


AEP 14 BUS TEST SYSTEM BUS CODE DIAGRAM

From a Transmission Network to a Graph

[University of Washington, 1999]

Graph G = (V, E)

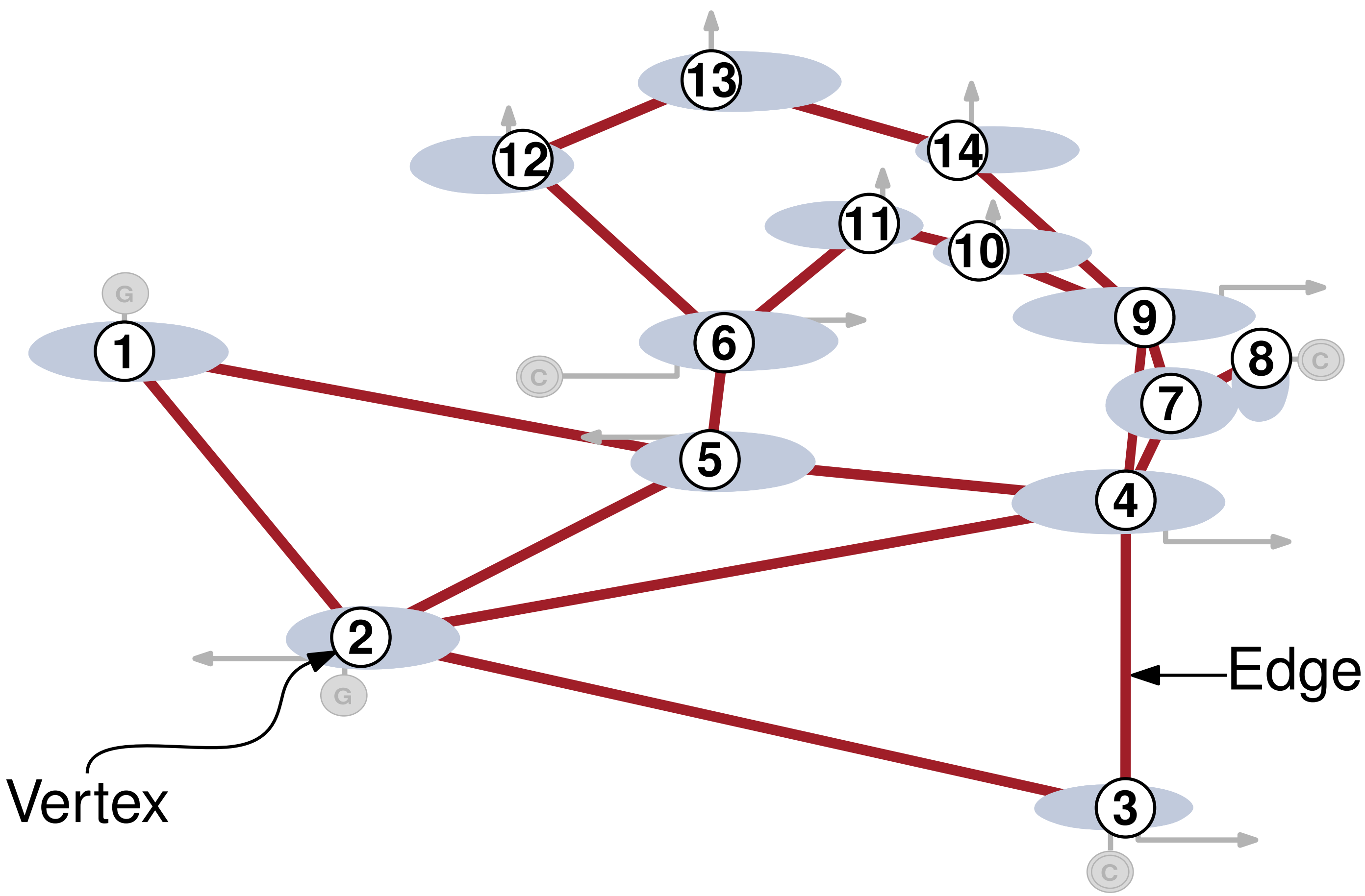


AEP 14 BUS TEST SYSTEM BUS CODE DIAGRAM

From a Transmission Network to a Graph

[University of Washington, 1999]

Graph $G = (V, E)$

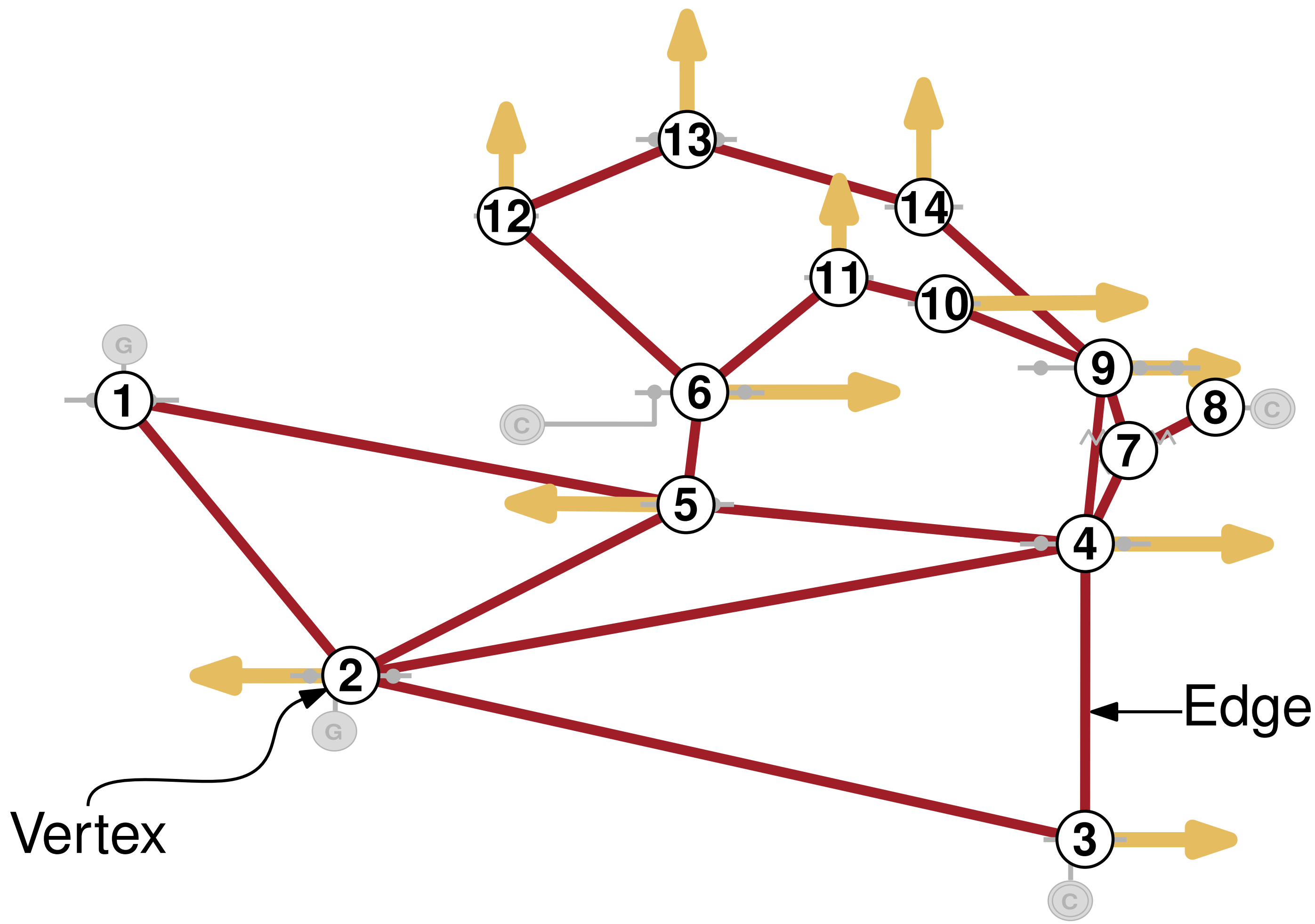


AEP 14 BUS TEST SYSTEM BUS CODE DIAGRAM

From a Transmission Network to a Graph

[University of Washington, 1999]

Graph G = (*V*, *E*)

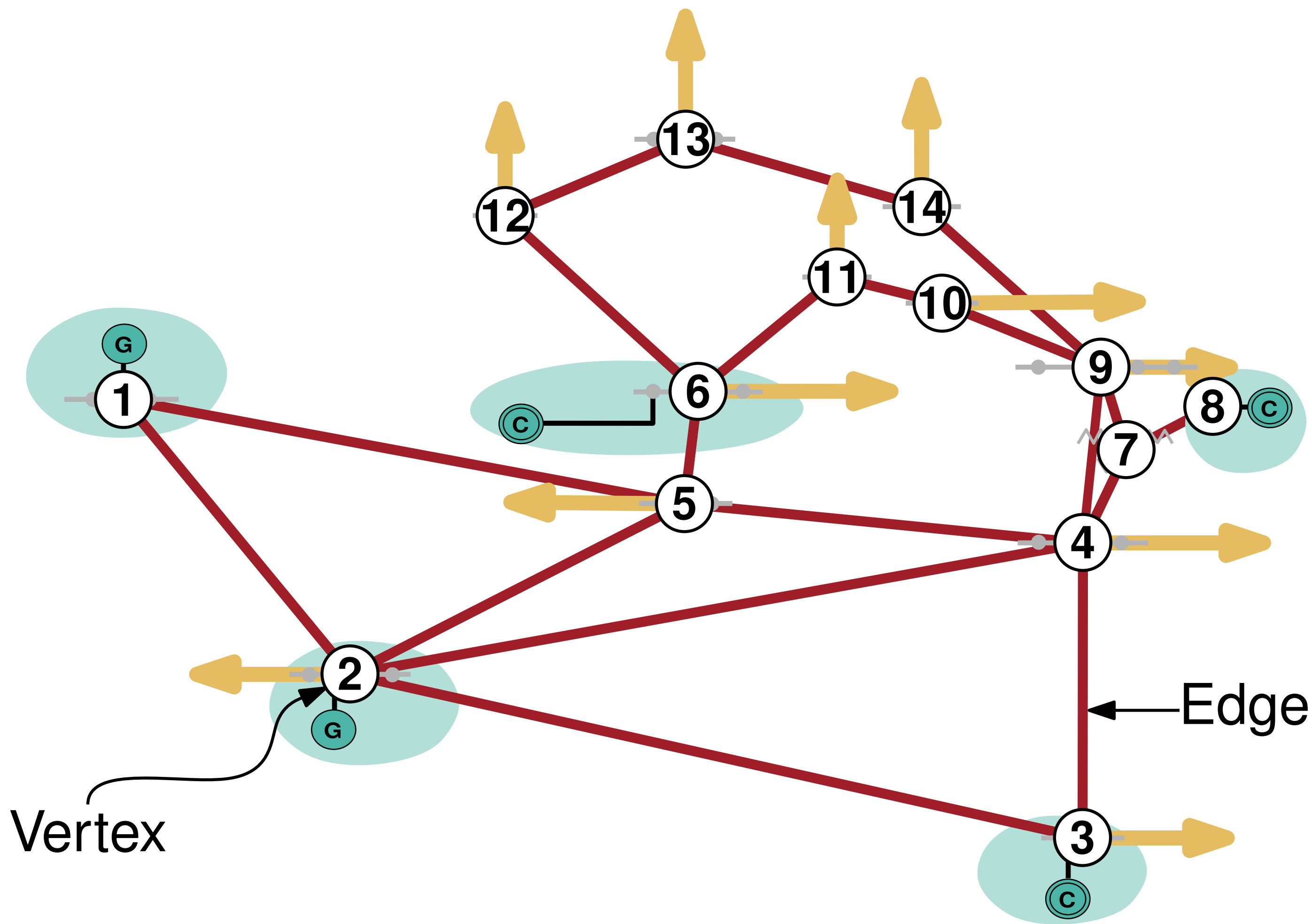


AEP 14 BUS TEST SYSTEM BUS CODE DIAGRAM

From a Transmission Network to a Graph

[University of Washington, 1999]

Graph G = (*V*, *E*)

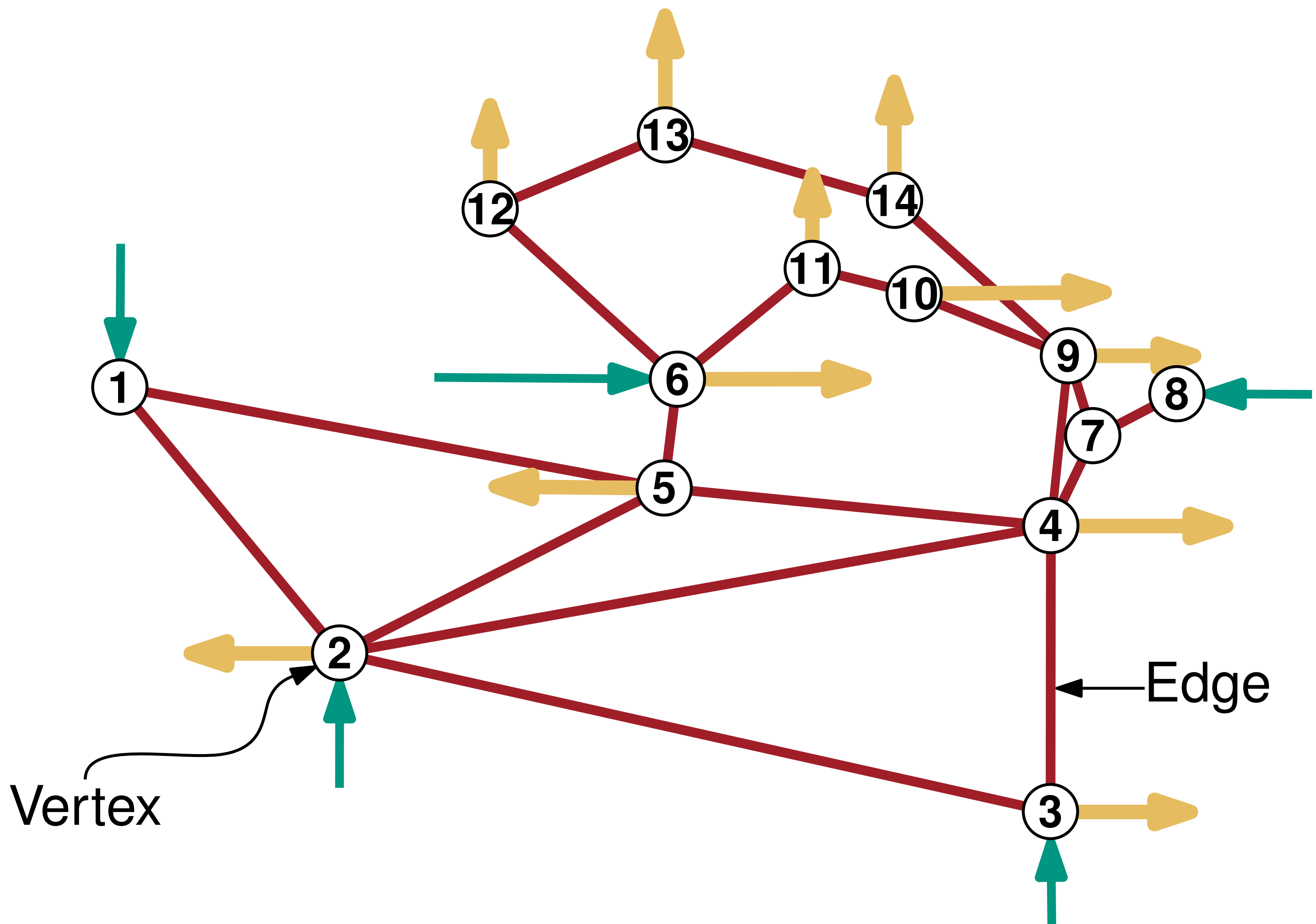


AEP 14 BUS TEST SYSTEM BUS CODE DIAGRAM

From a Transmission Network to a Graph

[University of Washington, 1999]

Graph G = (*V*, *E*)



AEP 14 BUS TEST SYSTEM BUS CODE DIAGRAM

Institute of Theoretical Informatics
Algorithmics Group

Input

- Let $G = (V, E)$ be the bidirected graph and $\underline{G} = (V, \underline{E})$ the underlying undirected graph.
- Set of vertices V (also called buses) with generators $V_G \subseteq V$, consumers $V_C \subseteq V \setminus V_G$, and intermediate vertices $V \setminus (V_G \cup V_C)$
- We denote by $\underline{E}$ the underlying undirected edge set with $\underline{e} \in \underline{E}$ such that $\underline{(u, v)} = \underline{(v, u)}$
- Network $\mathcal{N} = (G, V_G, V_C, \mathrm{cap}, b, \underline{d})$

  thermal line limits cap: $E \to \mathbb{R}_{\geq 0}$,

  susceptance b: $E \to \mathbb{R}_{\geq 0}$,

  demands lower bound $\underline{d}$: $V_C \to \mathbb{R}_{\geq 0}$.

Institute of Theoretical Informatics
Algorithmics Group

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

Institute of Theoretical Informatics
Algorithmics Group

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]

$\rightarrow$ Power grids are not easy.

$\rightarrow$ **Linearized AC** conservation of flow is **easy** to solve.

load capacity constraints
power flow constraints

Feasible Flow

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

$$
\begin{aligned}
f_{\text{net}}(u) &= 0 && \forall u \in V \setminus (V_G \cup V_C) \\
-\infty \leq f_{\text{net}}(u) &\leq -\underline{d} && \forall u \in V_C \\
0 \leq f_{\text{net}}(u) &\leq \infty && \forall u \in V_G
\end{aligned}
$$

- Flow f is called *feasible* if

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

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

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

Institute of Theoretical Informatics
Algorithmics Group

The Maximum Flow (MF) Problem

- 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) is denoted by $\text{MF}(\mathcal{N})$
$$\text{OPT}_{\text{MF}}(\mathcal{N}) = \max F(\mathcal{N}, f)$$
with f being a feasible flow meaning

$$\begin{aligned}
f_{\text{net}}(u) &= 0 & \forall u \in V \setminus (V_G \cup V_C) \\
-\infty \leq f_{\text{net}}(u) &\leq -\underline{d} & \forall u \in V_C \\
0 \leq f_{\text{net}}(u) &\leq \infty & \forall u \in V_G \\
|f(u, v)| &\leq \text{cap}(u, v) & \forall (u, v) \in E
\end{aligned}$$

Physical Feasible Flow

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

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

- 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 $\mathrm{MPF}(\mathcal{N})$ with value
$$\mathrm{OPT}_{\mathrm{MPF}}(\mathcal{N}) = \max F(\mathcal{N}, f)$$
with f being a *physical* feasible flow meaning

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

The Maximum Power Flow (MPF) Problem

- The Maximum Power Flow (MPF) is denoted by MPF($\mathcal{N}$) with value
$$\mathrm{OPT}_{\mathrm{MPF}}(\mathcal{N}) = \max F(\mathcal{N}, f)$$
with f being a *physical* feasible flow meaning

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

Institute of Theoretical Informatics
Algorithmics Group

The MAXIMUM TRANSMISSION SWITCHING FLOW (MTSF) Problem [Lehmann et al., 2014]

- The MAXIMUM TRANSMISSION SWITCHING FLOW (MTSF) is denoted by
$$\mathrm{MTSF}(\mathcal{N}) := \max_{S \subseteq E} \mathrm{MPF}(\mathcal{N} - S)$$
with value $\mathrm{OPT}_{\mathrm{MTSF}}(\mathcal{N})$ with f being a physical feasible flow meaning

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

Institute of Theoretical Informatics
Algorithmics Group

The MAXIMUM TRANSMISSION SWITCHING FLOW (MTSF) Problem [Lehmann et al., 2014]

- The MAXIMUM TRANSMISSION SWITCHING FLOW (MTSF) is denoted by
$$\mathrm{MTSF}(\mathcal{N}) := \max_{S \subseteq E} \mathrm{MPF}(\mathcal{N} - S)$$
with value $\mathrm{OPT}_{\mathrm{MTSF}}(\mathcal{N})$ with f being a physical feasible flow meaning

$$
\begin{aligned}
f_{\mathrm{net}}(u) &= 0 && \forall u \in V \setminus (V_G \cup V_C)\\
-\infty \leq f_{\mathrm{net}}(u) &\leq -\underline{d}_u && \forall u \in V_C\\
0 \leq f_{\mathrm{net}}(u) &\leq \infty && \forall u \in V_G\\
|f(u,v)| &\leq z(u,v) \cdot \mathrm{cap}(u,v) && \forall (u,v) \in E\\
b(u,v) \cdot \big(\theta(v) - \theta(u)\big) + \big(1 - z(u,v)\big) M &\geq f(u,v) && \forall (u,v) \in E\\
b(u,v) \cdot \big(\theta(v) - \theta(u)\big) - \big(1 - z(u,v)\big) M &\leq f(u,v) && \forall (u,v) \in E\\
\theta_{\min}(u) \leq \theta(u) &\leq \theta_{\max}(u) && \forall u \in V
\end{aligned}
$$

Institute of Theoretical Informatics
Algorithmics Group

The Maximum Transmission Switching Flow (MTSF) Problem [Lehmann et al., 2014]

Optimization Problem MTSF

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

Objective: Find a set $S \subseteq \underline{E}$ of switched edges such that $\mathrm{OPT}_{\mathrm{MPF}}(\mathcal{N} - S)$ 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 $S \subseteq \underline{E}$ such that there is an physical feasible flow f in $\mathcal{N} - S$ with flow value $F(\mathcal{N} - S, f) \geq k$?

Institute of Theoretical Informatics
Algorithmics Group

The OPTIMAL POWER FLOW (OPF) Problem

[Zimmerman et al., 2011]

- Network $\mathcal{N} = (G, V_G, V_C, \mathrm{cap}, b, \underline{d})$

- The OPTIMAL POWER FLOW (OPF) is denoted by OPF($\mathcal{N}$) with value
$$\mathrm{OPT}_{\mathrm{OPF}}(\mathcal{N}) := \min \textstyle\sum_{u \in V_G} \gamma_u \big(f_{\mathrm{net}}(u)\big)$$
with f being a physical feasible flow meaning

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

Institute of Theoretical Informatics
Algorithmics Group

The OPTIMAL POWER FLOW (OPF) Problem

[Zimmerman et al., 2011]

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

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

with f being a physical feasible flow meaning

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

Institute of Theoretical Informatics
Algorithmics Group

The OPTIMAL POWER FLOW (OPF) Problem

[Zimmerman et al., 2011]

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

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

  with f being a physical feasible flow meaning

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

Institute of Theoretical Informatics
Algorithmics Group

The OPTIMAL POWER FLOW (OPF) Problem

[Zimmerman et al., 2011]

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

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

with f being a physical feasible flow meaning

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

Institute of Theoretical Informatics
Algorithmics Group

The Optimal Transmission Switching (OTS) Problem [Fisher et al., 2008]

- The Optimal Transmission Switching (OTS) is denoted by

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

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

$$\begin{aligned}
f_{\mathrm{net}}(u) &= 0 && \forall u \in V \setminus (V_G \cup V_C)\\
f_{\mathrm{net}}(u) &= -d_u && \forall u \in V_C\\
\underline{x} \leq f_{\mathrm{net}}(u) &\leq \overline{x} && \forall u \in V_G\\
|f(u,v)| &\leq z(u,v) \cdot \mathrm{cap}(u,v) && \forall (u,v) \in E\\
b(u,v) \cdot \big(\theta(v) - \theta(u)\big) + \big(1 - z(u,v)\big) M &\geq f(u,v) && \forall (u,v) \in E\\
b(u,v) \cdot \big(\theta(v) - \theta(u)\big) - \big(1 - z(u,v)\big) M &\leq f(u,v) && \forall (u,v) \in E\\
\theta_{\min}(u) \leq \theta(u) &\leq \theta_{\max}(u) && \forall u \in V
\end{aligned}$$

Institute of Theoretical Informatics Algorithmics Group

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 $S \subseteq \underline{E}$ of switched edges such that $\text{OPT}_{\text{OPF}}(\mathcal{N} - S)$ is minimum among all choices of switched edges 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 $S \subseteq \underline{E}$ such that there is an physical feasible flow f in $\mathcal{N} - S$ with cost $\sum_{u \in V_G} \gamma_u \big(f_{\text{net}}(u)\big) \leq k$?

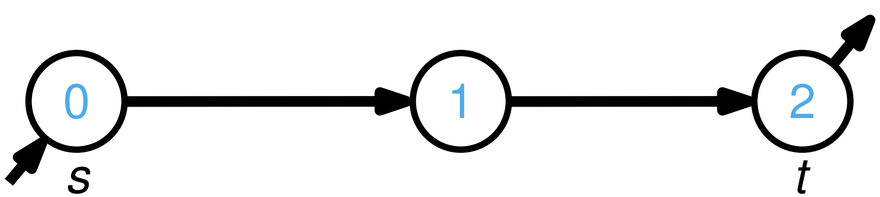
Connection to the DC-Model

- 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 $\Delta\theta$ behave like the voltages U

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

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

Series Circuit

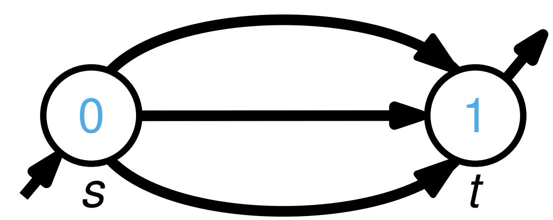


$$B_{\text{tot}} = b(s,t) = \frac{1}{\sum_{k=1}^{n} \frac{1}{b_k}} = \frac{1}{2}$$

$$f(s,t) = 1$$

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

Parallel Circuit



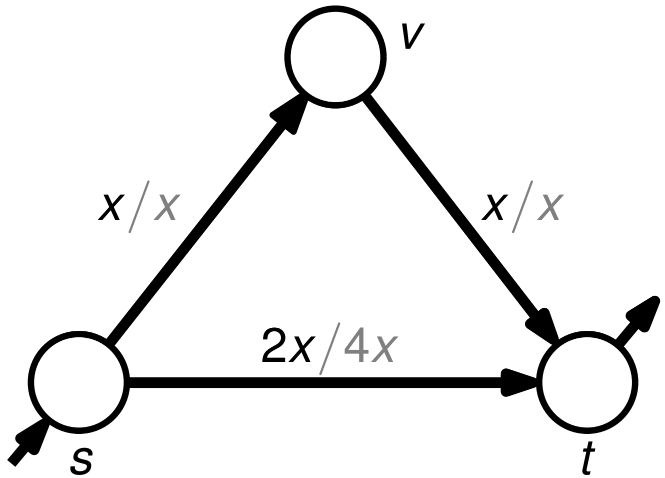
$$B_{\text{tot}} = b(s,t) = \sum_{k=1}^{n} b_k = 3$$

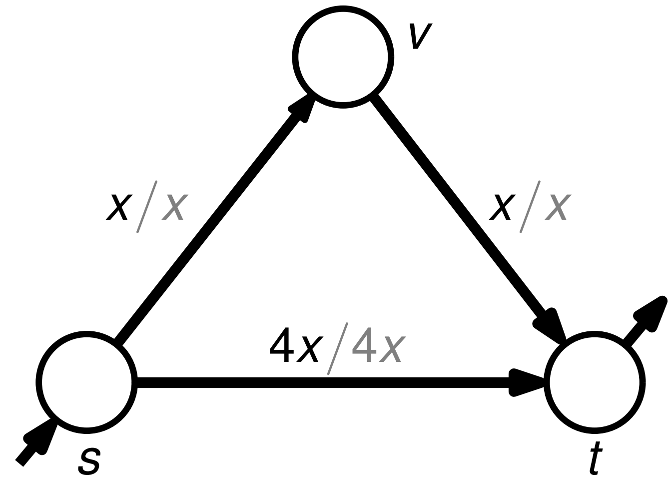
$$f(s,t) = 3$$

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

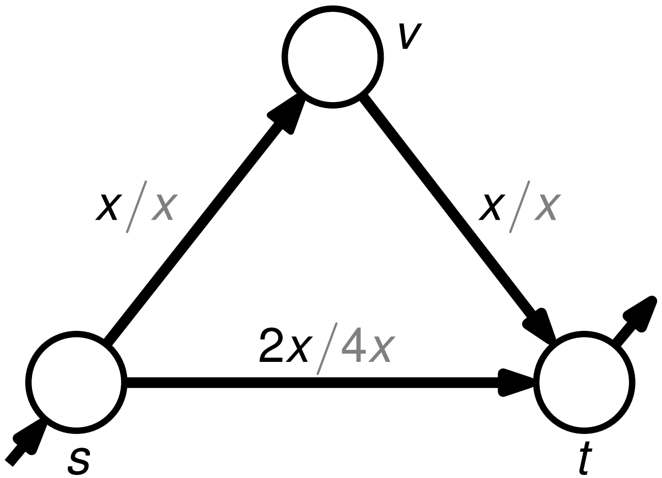
Institute of Theoretical Informatics
Algorithmics Group

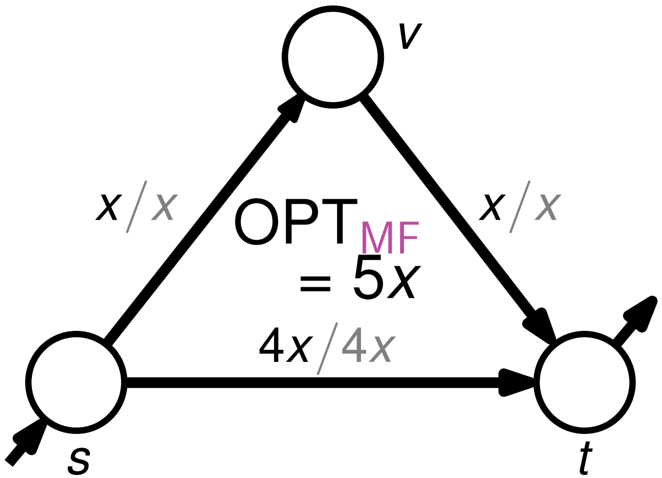
The MTSF Problem





The MTSF Problem



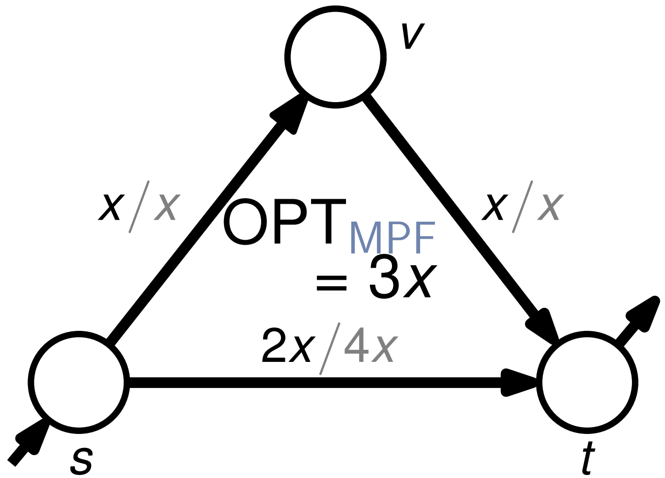


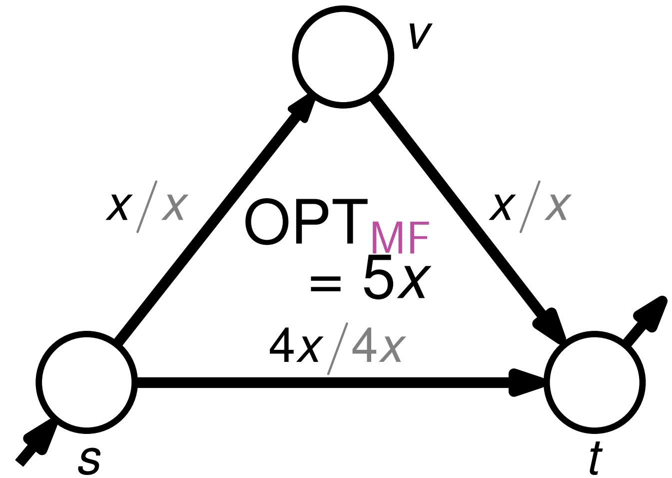
flow model

upper bound

Institute of Theoretical Informatics
Algorithmics Group

The MTSF Problem





physical model
(AC linearization)

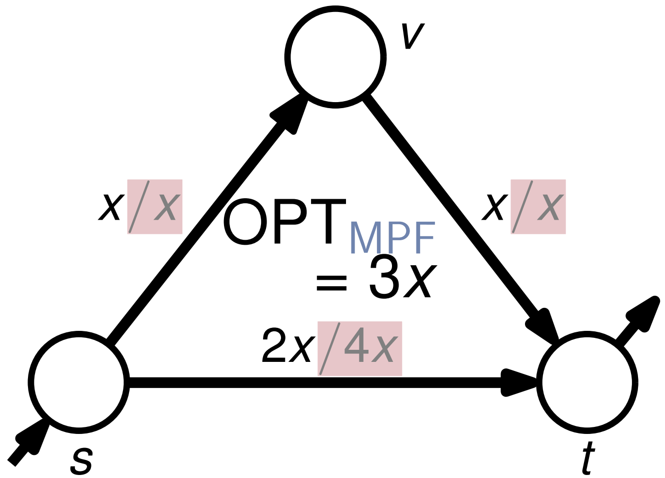
lower bound

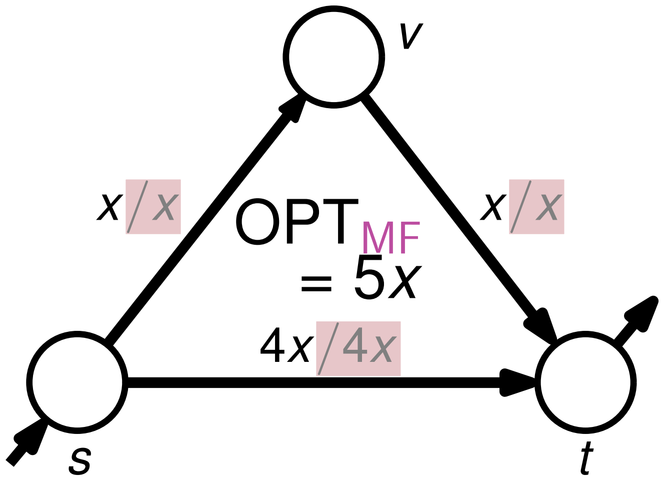
flow model

upper bound

Institute of Theoretical Informatics
Algorithmics Group

The MTSF Problem





physical model
(AC linearization)

lower bound

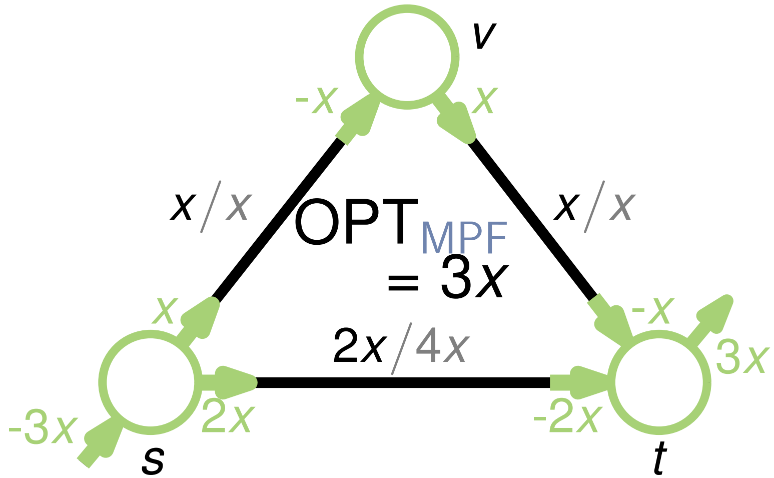
capacity constraints

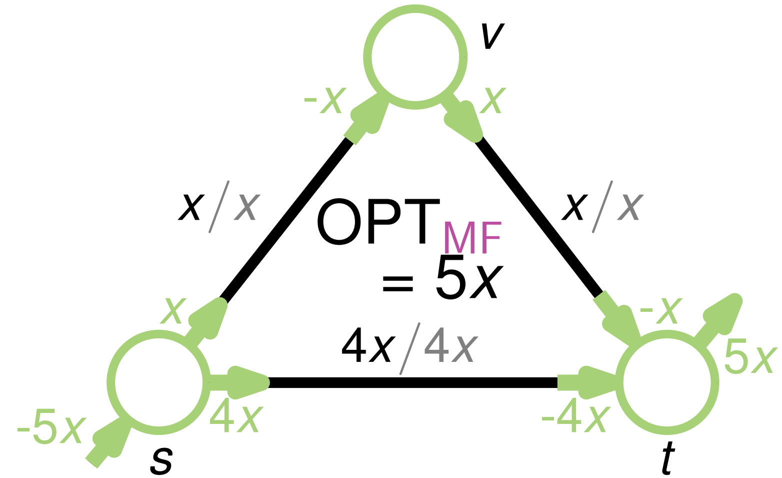
flow model

upper bound

Institute of Theoretical Informatics
Algorithmics Group

The MTSF Problem





physical model
(AC linearization)

flow model

lower bound

upper bound

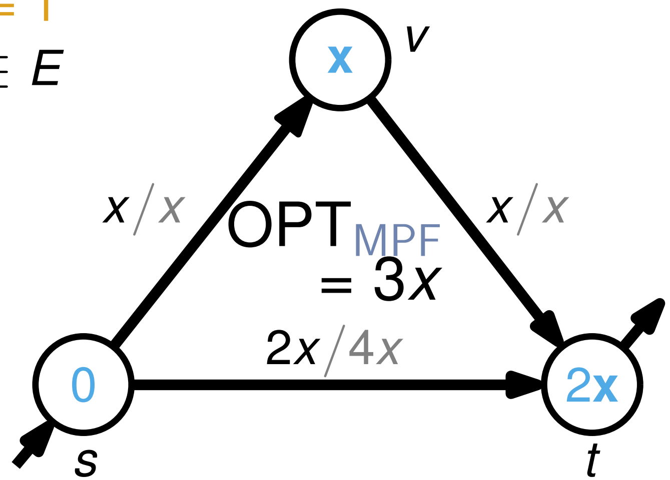
capacity constraints

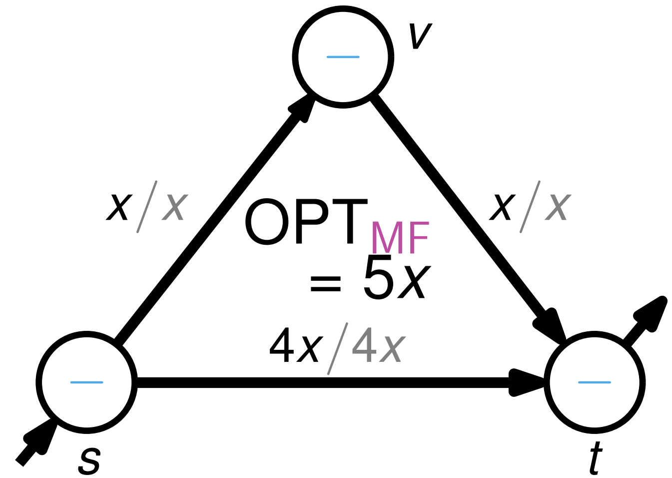
Kirchhoff's Current Law (KCL)

Institute of Theoretical Informatics
Algorithmics Group

The MTSF Problem

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





physical model
(AC linearization)

lower bound

flow model

upper bound

capacity constraints

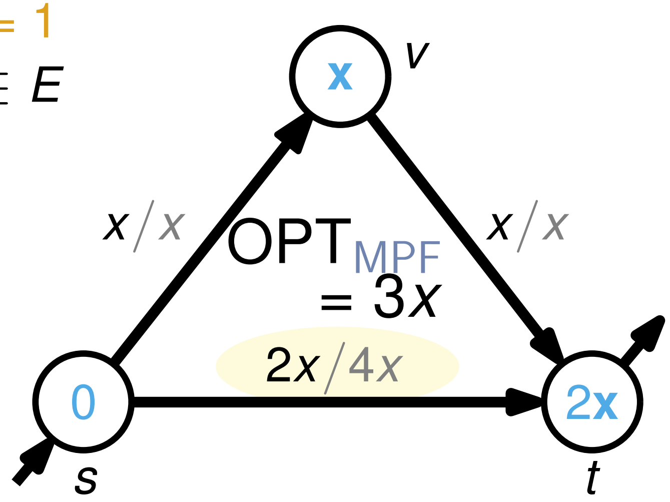
Kirchhoff's Current Law (KCL)

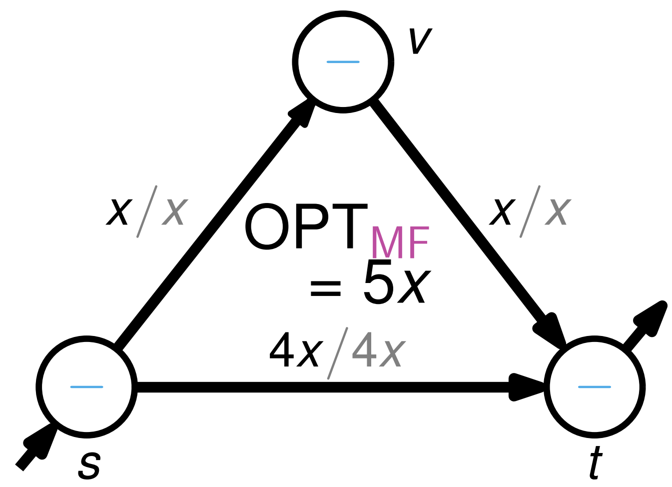
DC power flow constraints

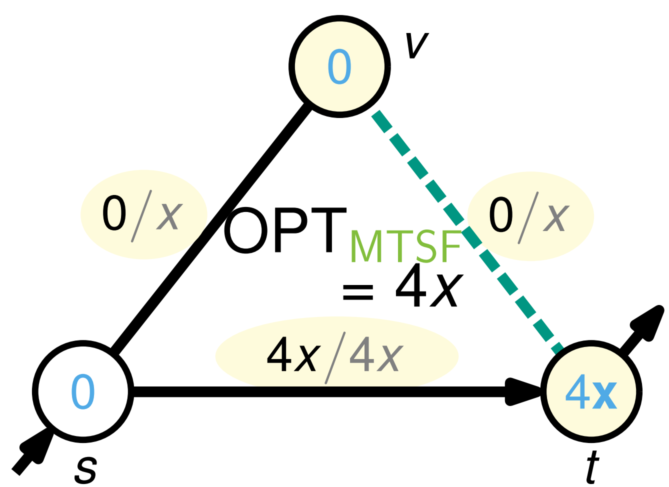
$$\forall (u, v) \in E\colon\ f(u, v) = b(u, v)\big(\theta(v) - \theta(u)\big)$$

The MTSF Problem

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





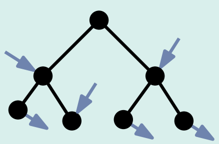


$$\forall (u, v) \in E\colon\ f(u, v) = b(u, v)\big(\theta(v) - \theta(u)\big)$$

Institute of Theoretical Informatics
Algorithmics Group

The MTSF Problem

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



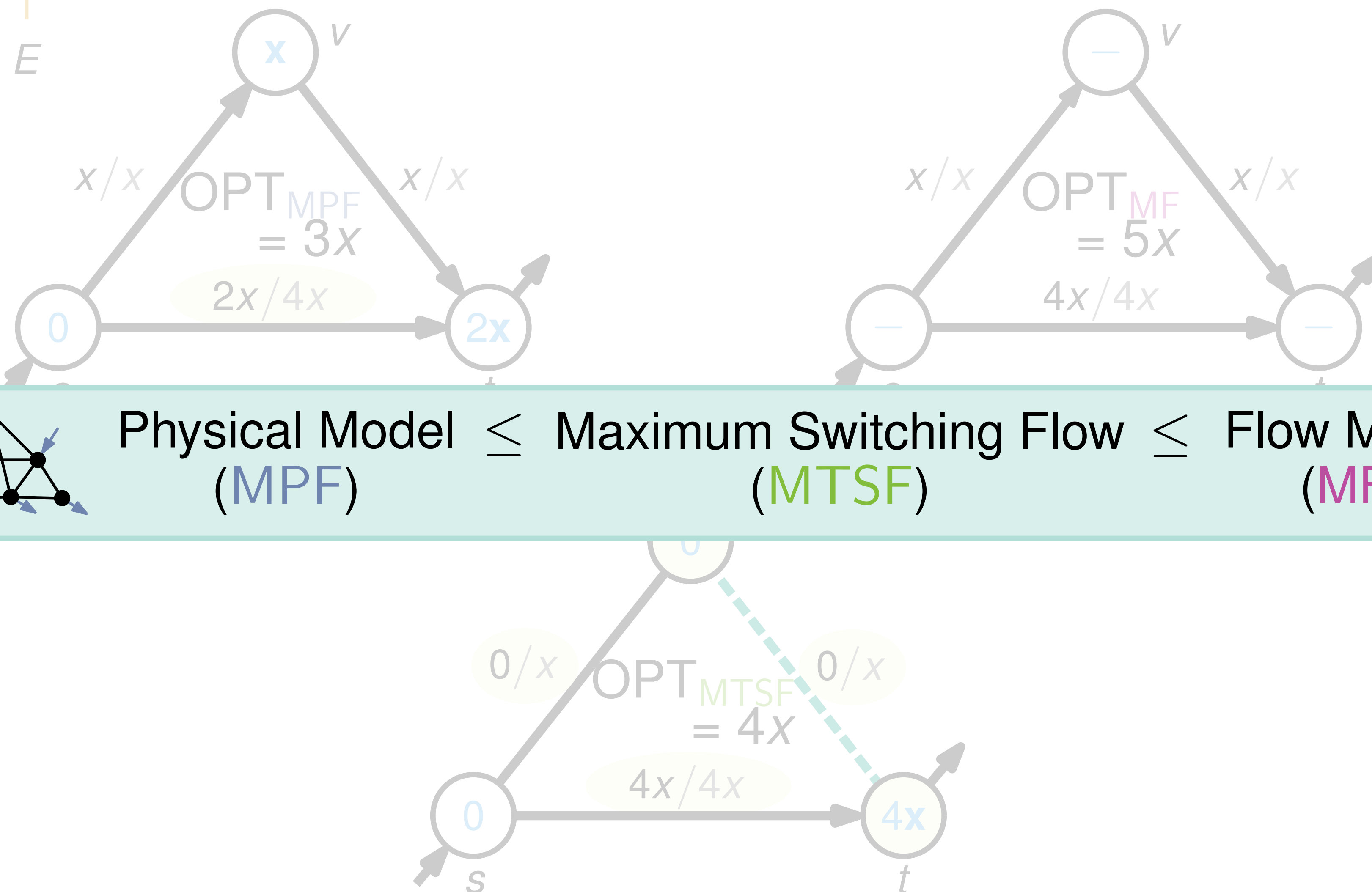
Physical Model (MPF) = Maximum Switching Flow (MTSF) = Flow Model (MF)

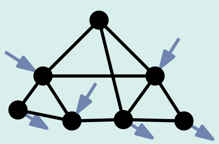
$$\forall (u,v) \in E: \; f(u,v) = b(u,v)\big(\theta(v) - \theta(u)\big)$$

Institute of Theoretical Informatics
Algorithmics Group

The MTSF Problem

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





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

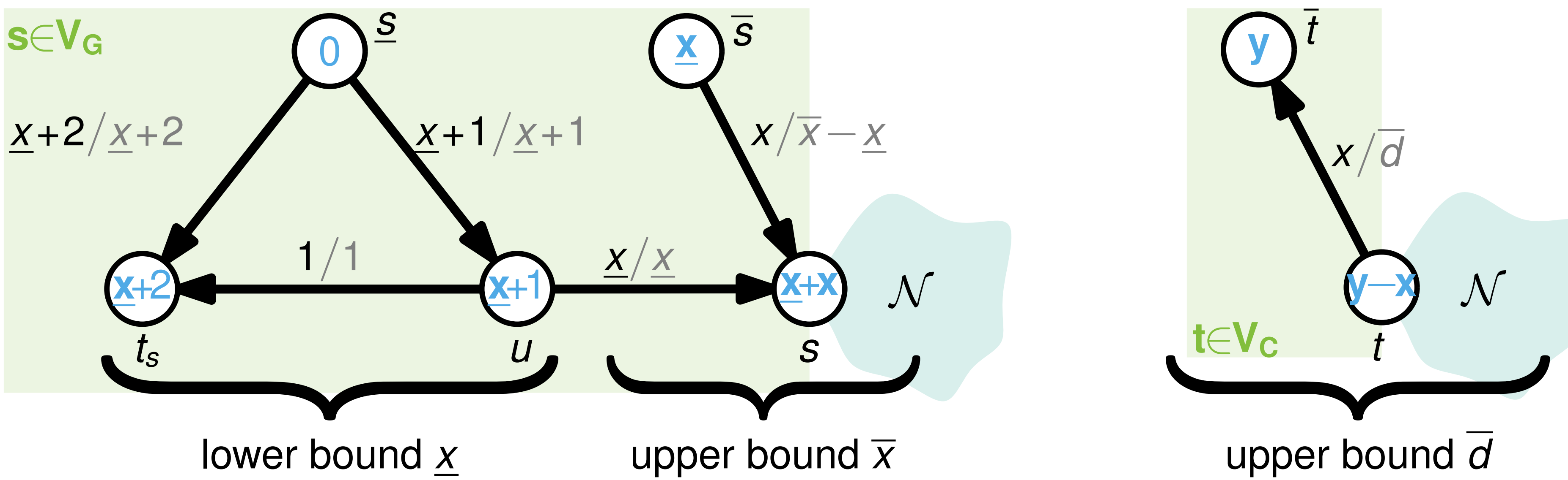
$$\forall (u, v) \in E: f(u, v) = b(u, v)\big(\theta(v) - \theta(u)\big)$$

Institute of Theoretical Informatics
Algorithmics Group

Network Modeling – Bounded to Unbounded

- Transformation from a bounded network $\mathcal{N}_{\text{bounded}} = (G, V_G, V_C, \text{cap}, b, \underline{x}, \overline{x}, \underline{d}, \overline{d})$ to an unbounded network $\mathcal{N} = (G, V_G, V_C, \text{cap}, b, \underline{d})$

Model Generator Bounds in $\mathcal{N}$ **Model Demand Bounds in $\mathcal{N}$**



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

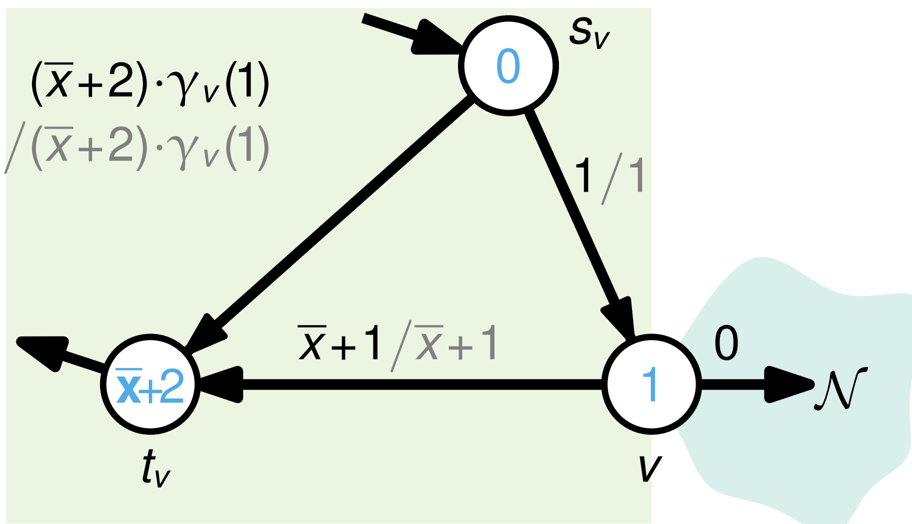
Every bounded MTSF can be transformed into an unbounded MTSF on a network with size linear in $|V|$ and $|E|$.

Institute of Theoretical Informatics
Algorithmics Group

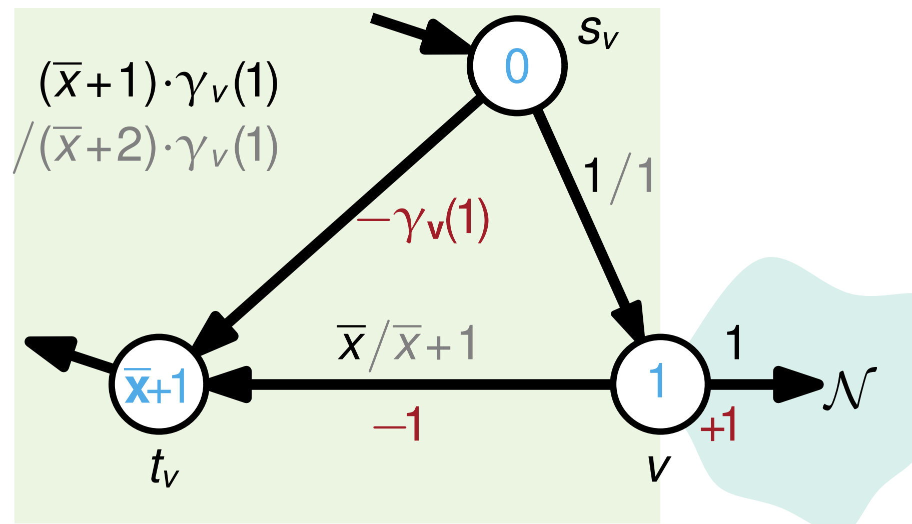
Network Modeling – OTS to MTSF

- OTS-instance $\mathcal{N} = (G, V_G, V_C, \text{cap}, b, \underline{x}, \overline{x}, d)$
- $\gamma_v(1)$ 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 $\mathcal{N}$

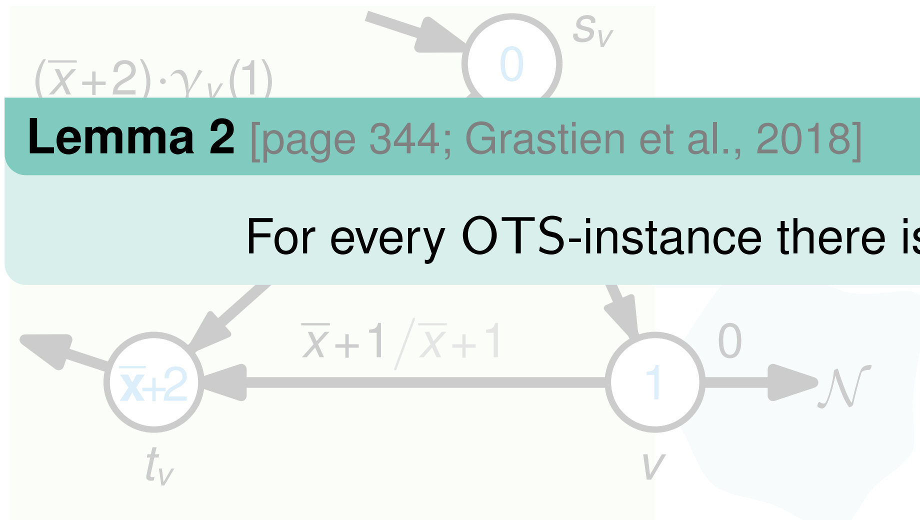


- Feasible flow in $\mathcal{N}$ with cost k $\rightarrow$ feasible flow in $\mathcal{N}$' with flow value $M - k$
- $M = \sum_{v \in V_G} \left((\overline{x} + 2) \cdot \gamma_v(1) + \overline{x}_v + 1 \right)$

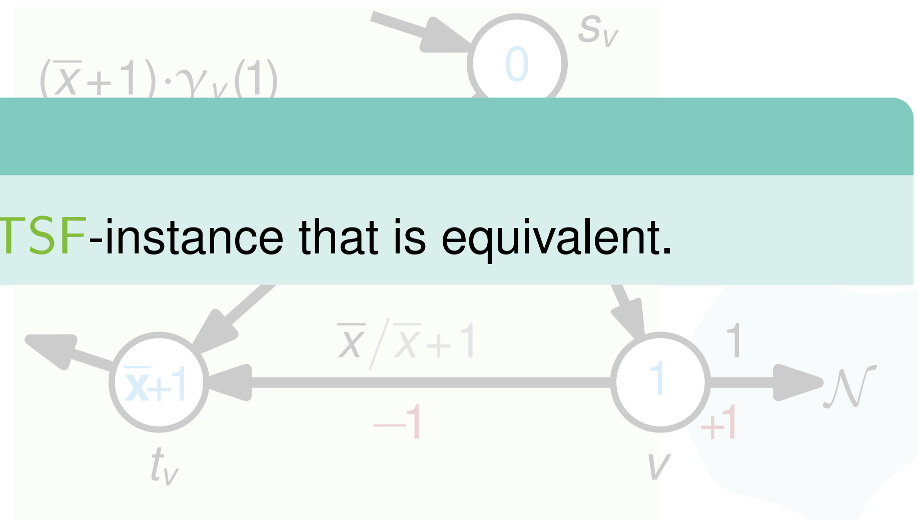
Network Modeling – OTS to MTSF

- OTS-instance $\mathcal{N} = (G, V_G, V_C, \text{cap}, b, \underline{x}, \overline{x}, d)$
- $\gamma_v(1)$ 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.

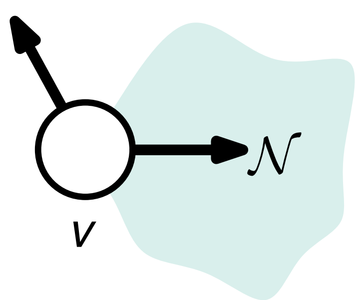
- Feasible flow in $\mathcal{N}$ with cost k $\rightarrow$ feasible flow in $\mathcal{N}$' with flow value $M - k$
- $M = \sum_{v \in V_G} \left((\overline{x} + 2) \cdot \gamma_v(1) + \overline{x}_v + 1 \right)$

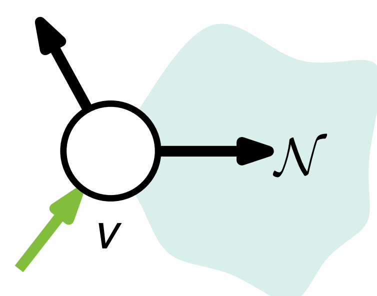
Network Modeling – MTSF to OTS

[Lehmann et al., 2014]

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

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





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

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

Institute of Theoretical Informatics
Algorithmics Group

Network Modeling – MTSF to OTS

[Lehmann et al., 2014]

- MTSF-instance $\mathcal{N} = (G, V_G, V_C, \mathrm{cap}, b, \overline{x}, \underline{d}, \overline{d})$ with $V_G \cap V_C = \emptyset$
- OTS-instance $\mathcal{N}' = (G, V'_G, V_C, \mathrm{cap}, b, \overline{x}', d' = \underline{d}' = \overline{d}, \gamma)$
- The new generator set $V'_G = V_G \cup \{v \in V_C | \overline{d}_v > 0\}$
- $\gamma_v(1) = 1$ 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.

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

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

Network Modeling – MTSF to OTS

[Lehmann et al., 2014]

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

$$\sum_{v \in V} \overline{d}_v = \mathrm{MTSF}(\mathcal{N}) = \mathrm{OTS}(\mathcal{N}')$$

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

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

Institute of Theoretical Informatics
Algorithmics Group

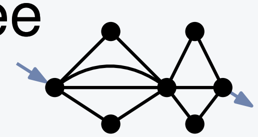
Overview of the MTSF Results

Graph Structure	Complexity	Algorithm

Overview of the MTSF Results

easy

one generator, one load

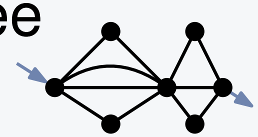
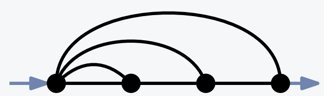
Graph Structure	Complexity	Algorithm
penrose-minor-free graphs 	polynomial-time solvable	DTP

Institute of Theoretical Informatics
Algorithmics Group

Overview of the MTSF Results

complexity

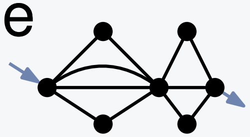
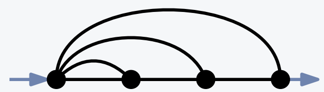
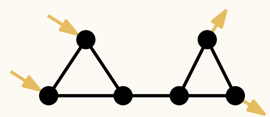
one generator, one load

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

Institute of Theoretical Informatics
Algorithmics Group

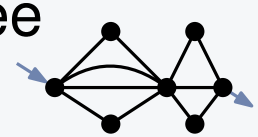
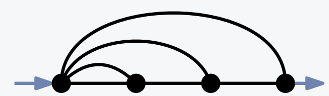
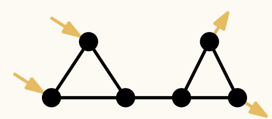
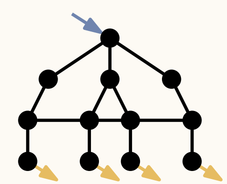
Overview of the MTSF Results

complexity

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

Institute of Theoretical Informatics
Algorithmics Group

Overview of the MTSF Results

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

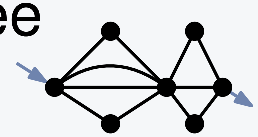
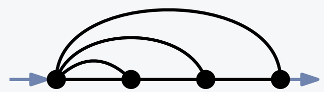
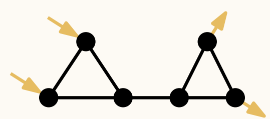
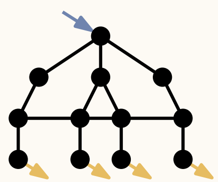
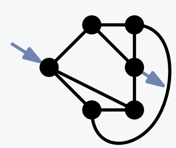
complexity

Institute of Theoretical Informatics
Algorithmics Group

Overview of the MTSF Results

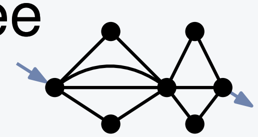
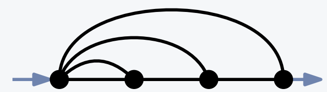
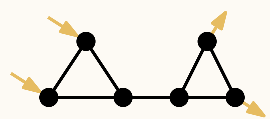
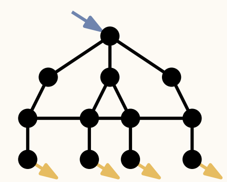
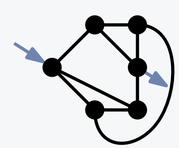
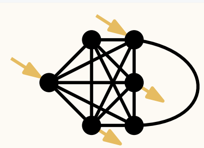
Karlsruhe Institute of Technology

complexity

	Graph Structure		Complexity	Algorithm
one generator, one load	penrose-minor-free graphs		polynomial-time solvable	DTP ✓
	series-parallel graphs		NP-hard	✗
arbitrary generators, arbitrary loads	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]	✗

Overview of the MTSF Results

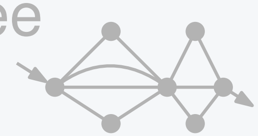
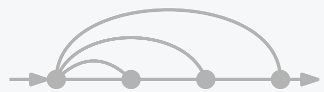
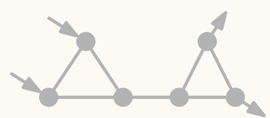
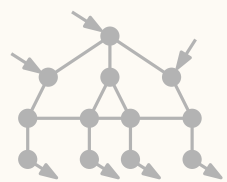
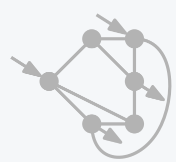
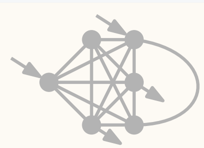
complexity

	Graph Structure		Complexity	Algorithm
one generator, one load	penrose-minor-free graphs		polynomial-time solvable	DTP ✓
	series-parallel graphs		NP-hard	✗
arbitrary generators, arbitrary loads	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]	✗
$\lvert V_G\rvert=2$, $\lvert V_C\rvert=2$	arbitrary graphs		non-APX [Lehmann et al., 2014]	✗

Institute of Theoretical Informatics
Algorithmics Group

Overview of the MTSF Results

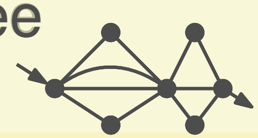
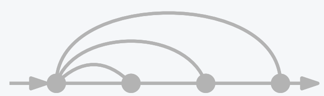
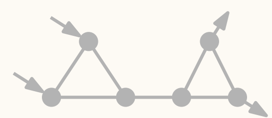
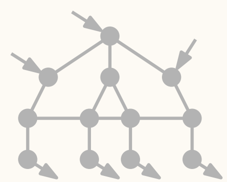
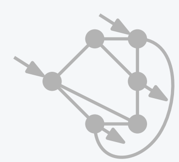
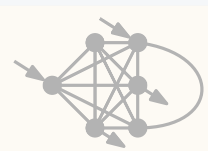
complexity

	Graph Structure		Complexity	Algorithm
one generator, one load	penrose-minor-free graphs		polynomial-time solvable	DTP ✓
	series-parallel graphs		NP-hard	✗
arbitrary generators, arbitrary loads	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]	✗
$\lvert V_G \rvert = 2$, $\lvert V_C \rvert = 2$	arbitrary graphs		non-APX [Lehmann et al., 2014]	✗

Institute of Theoretical Informatics
Algorithmics Group

Overview of the MTSF Results

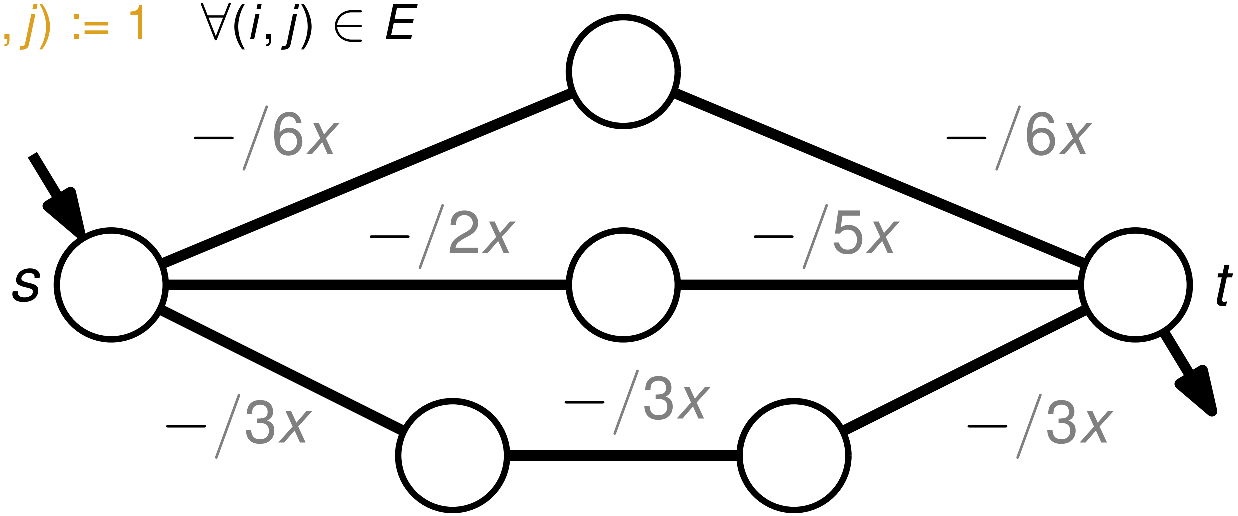
Karlsruhe Institute of Technology

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

Institute of Theoretical Informatics
Algorithmics Group

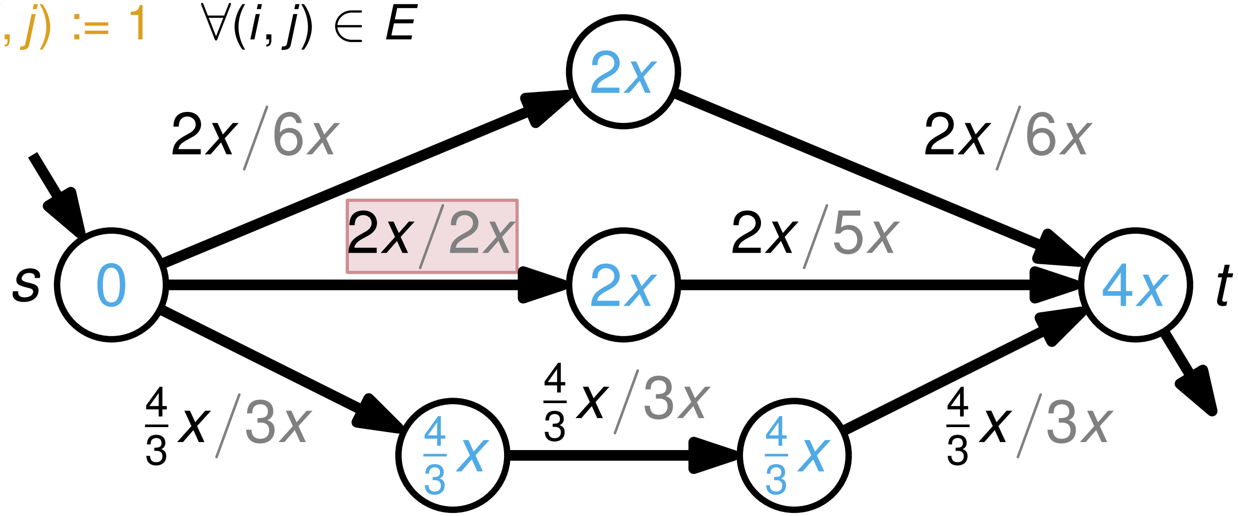
Switching on Parallel Paths

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



Switching on Parallel Paths

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



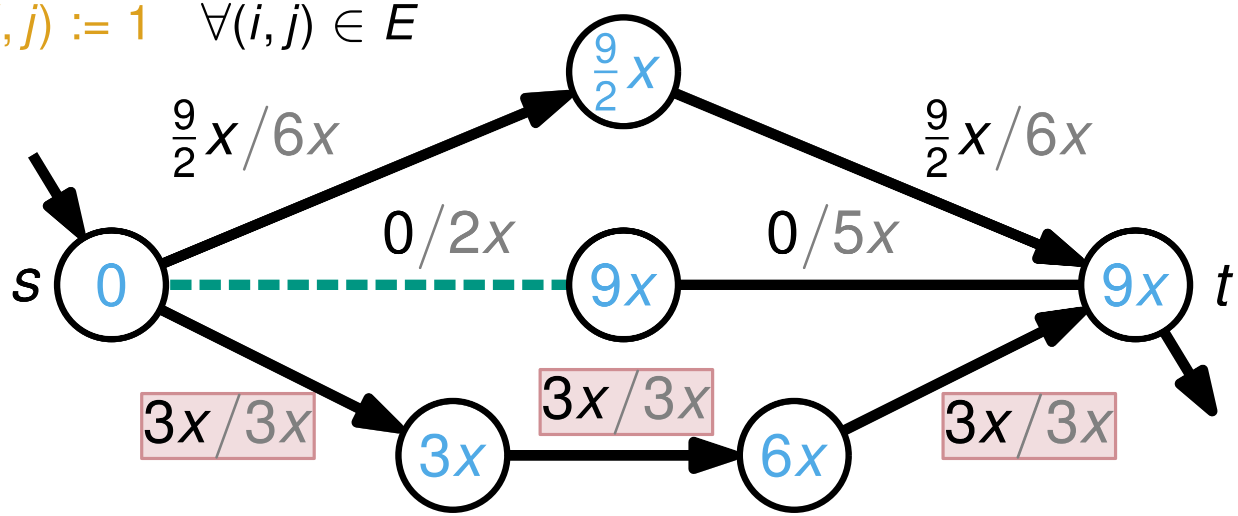
$$F = \frac{16}{3}x \approx 5.33x$$

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

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

Switching on Parallel Paths

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



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

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

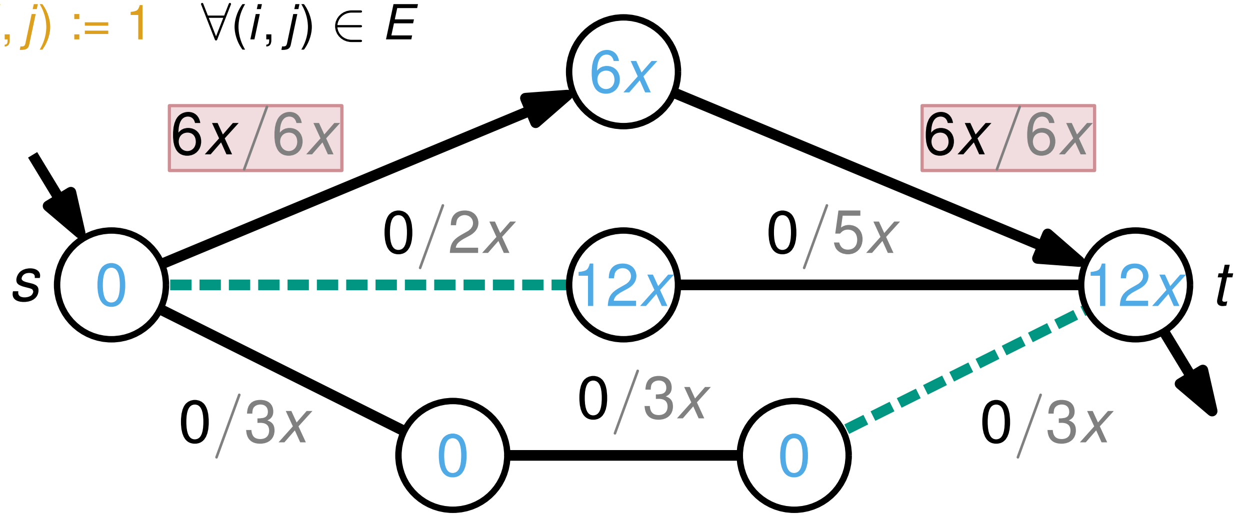
$\Delta\theta(\pi_2) = 2 \cdot 2x = 4x$

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

Institute of Theoretical Informatics
Algorithmics Group

Switching on Parallel Paths

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



$F = 6x$

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

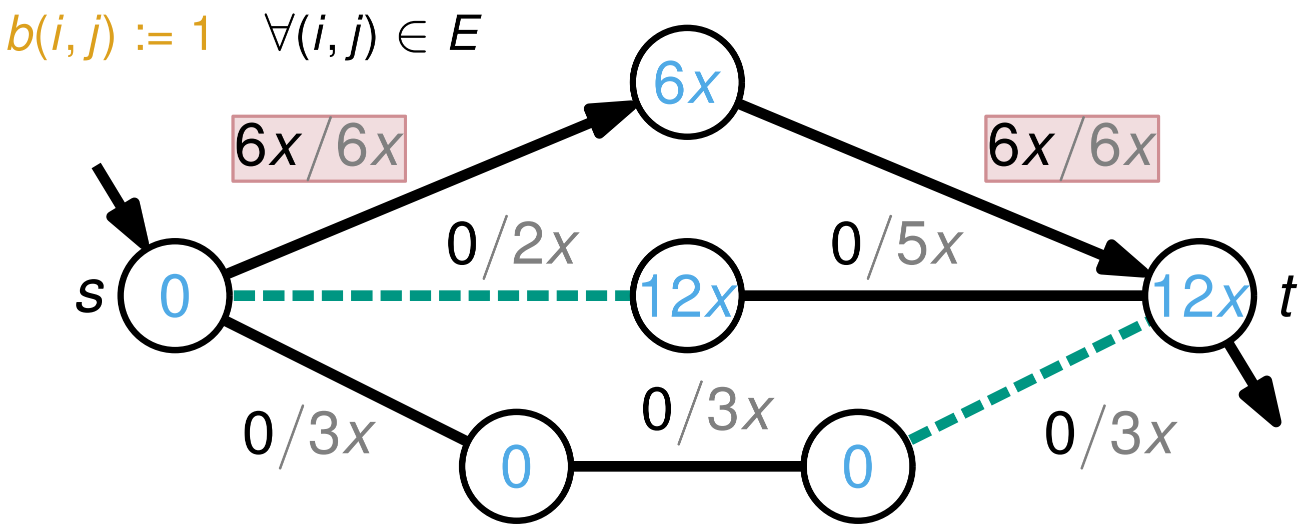
$\Delta\theta(\pi_1) = 2 \cdot 6x = 12x$

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

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

Institute of Theoretical Informatics
Algorithmics Group

Switching on Parallel Paths



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

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

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

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

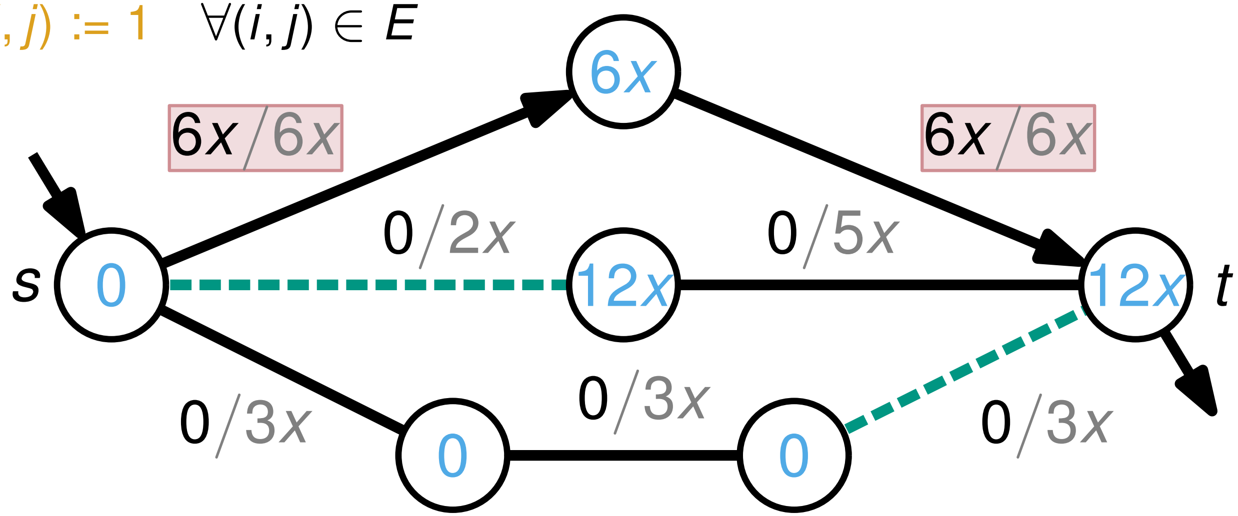
$F = 6x$

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

Institute of Theoretical Informatics
Algorithmics Group

Switching on Parallel Paths

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



$F = 6x$

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

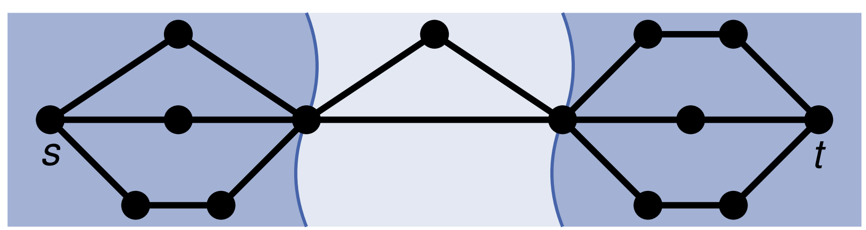
$\Delta\theta(\pi_1) = 2 \cdot 6x = 12x$

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

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

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

The same holds for series compositions of these graphs.



Institute of Theoretical Informatics
Algorithmics Group

Dominating Theta Path (DTP)

[Section 5; Grastien et al., 2018]

Karlsruhe Institute of Technology

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:

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

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:

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

Angle Difference of π:

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

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:

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

Angle Difference of π:

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

Dominating Theta Path (DTP):

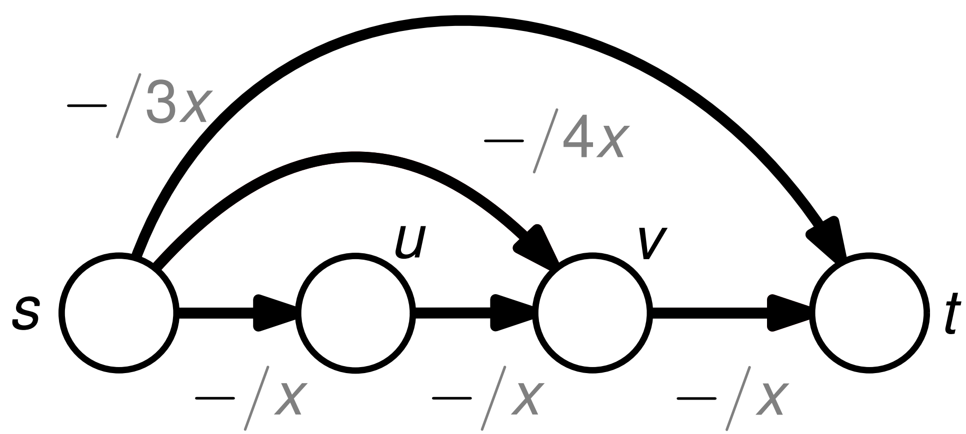
$$\Delta\theta_{\min}(u, v) := \min\{\Delta\theta(\pi) \mid \pi \text{ is a } u\text{-}v\text{-path}\}$$

Computing DTP [Section 5; Grastien et al., 2018]

Description:

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

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

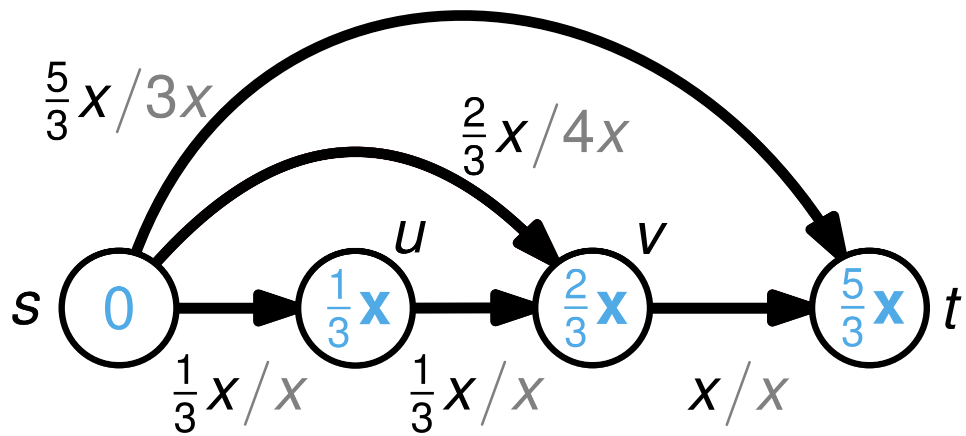


Computing DTP [Section 5; Grastien et al., 2018]

Description:

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

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



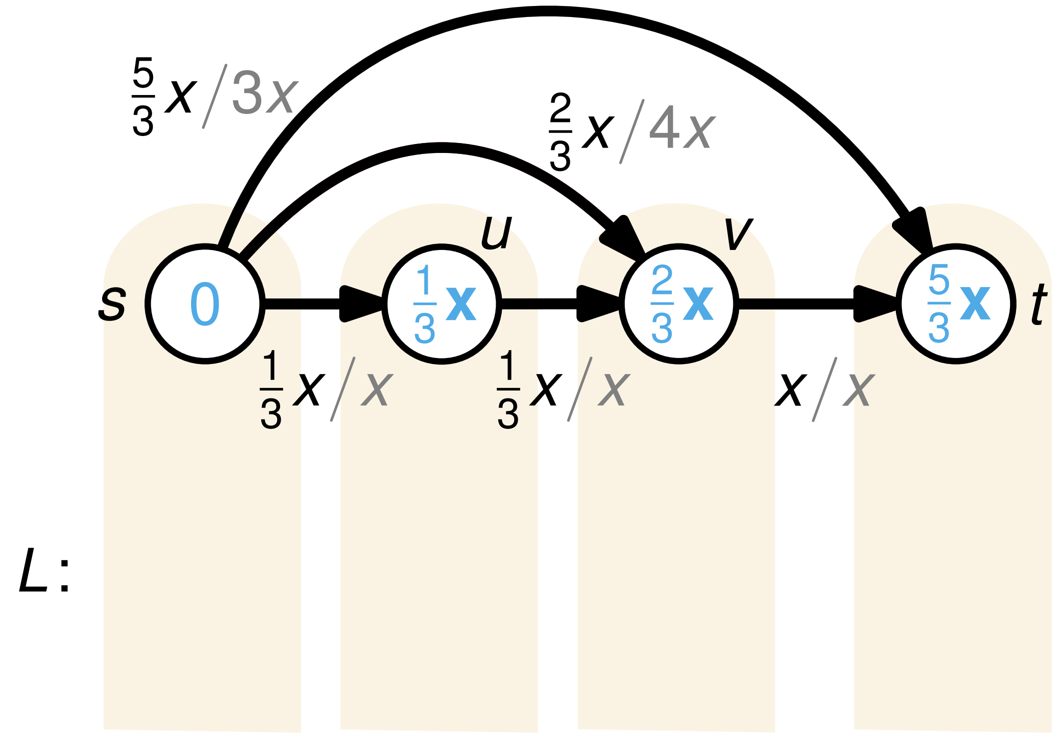
$\mathrm{OPT}_{\mathrm{MPF}} = \frac{8}{3}x$

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

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



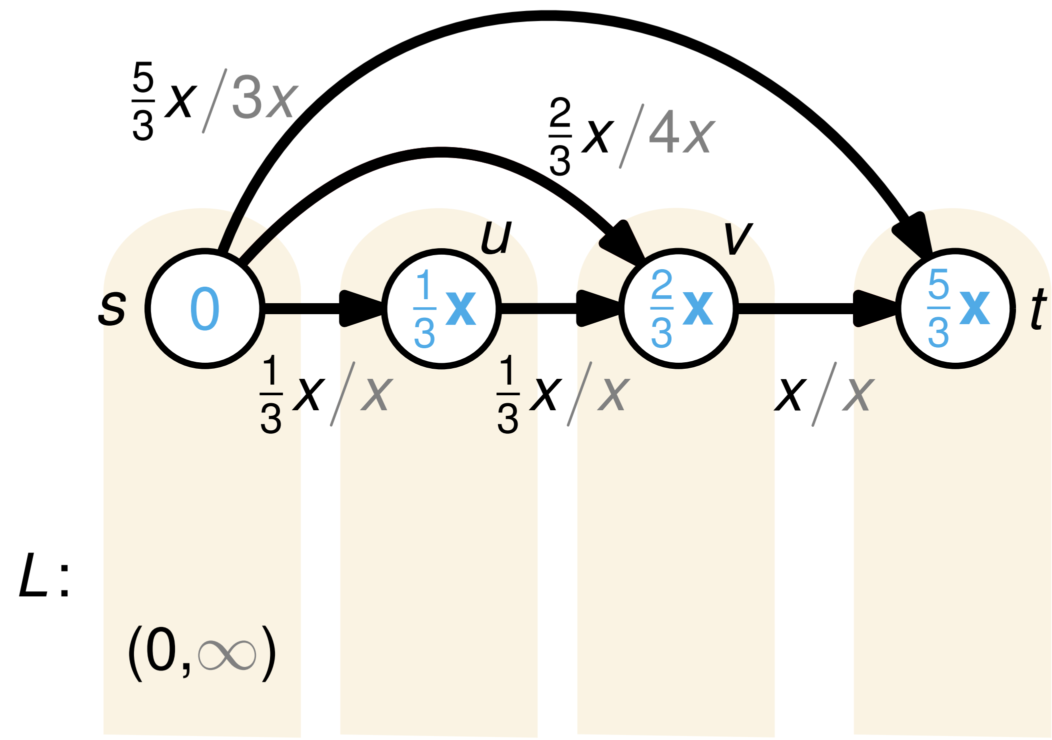
$\mathrm{OPT}_{\mathrm{MPF}} = \frac{8}{3}x$

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

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



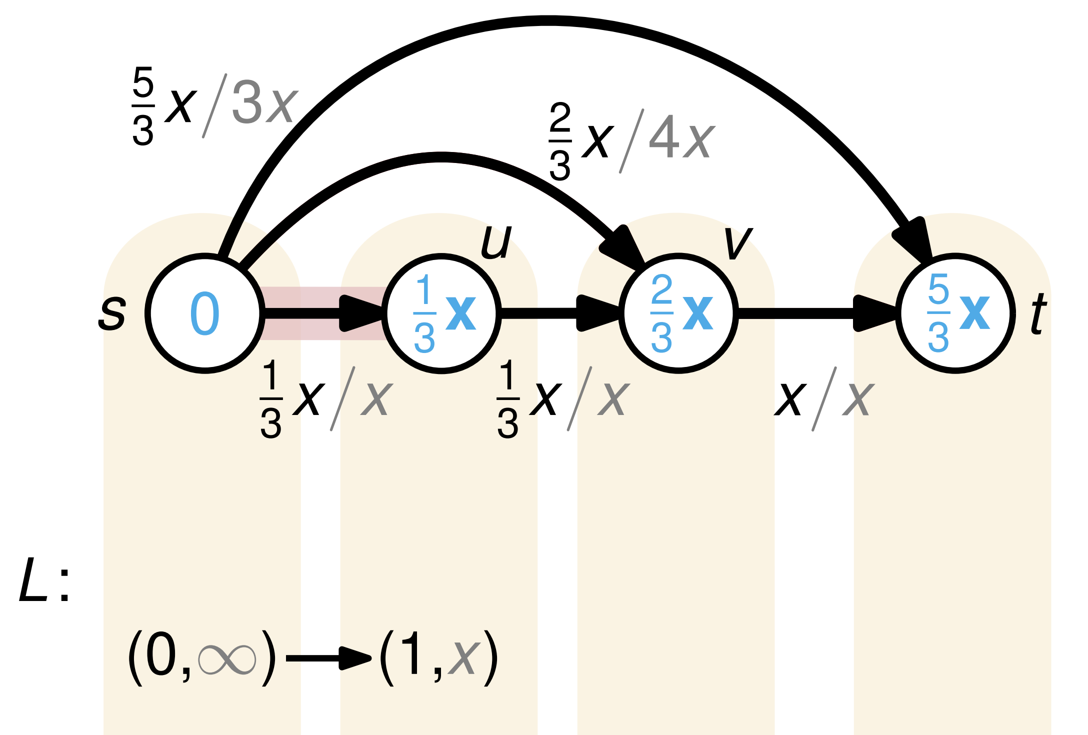
$\text{OPT}_{\text{MPF}} = \frac{8}{3}x$

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

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



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

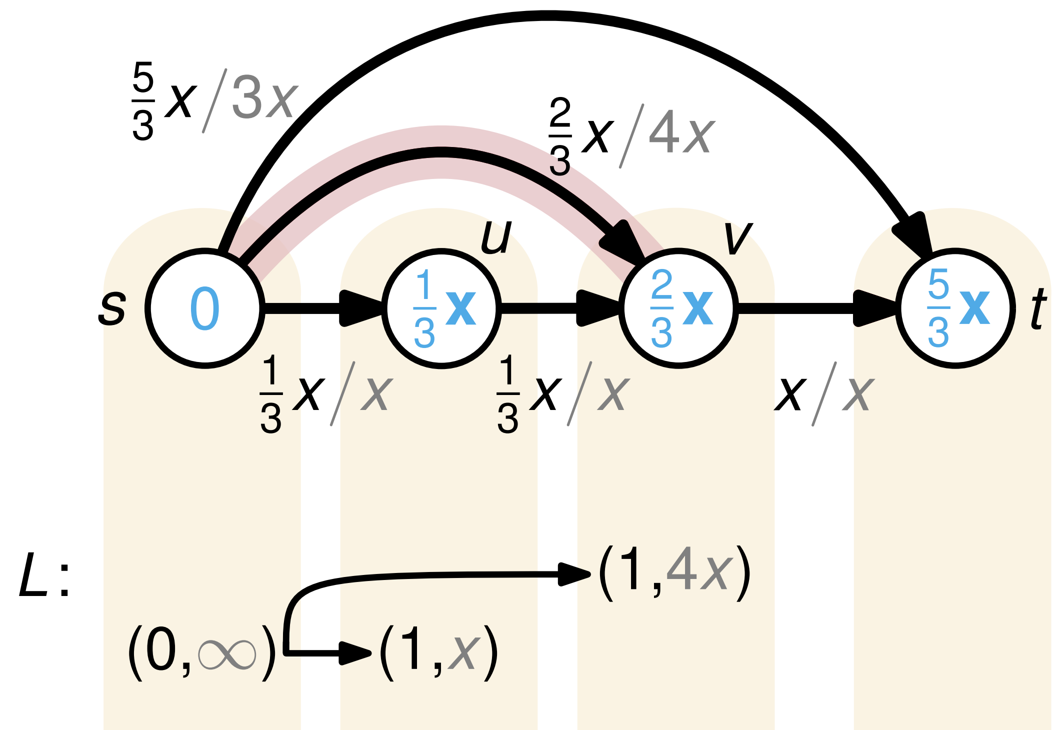
Institute of Theoretical Informatics
Algorithmics Group

Computing DTP [Section 5; Grastien et al., 2018]

Description:

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

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



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

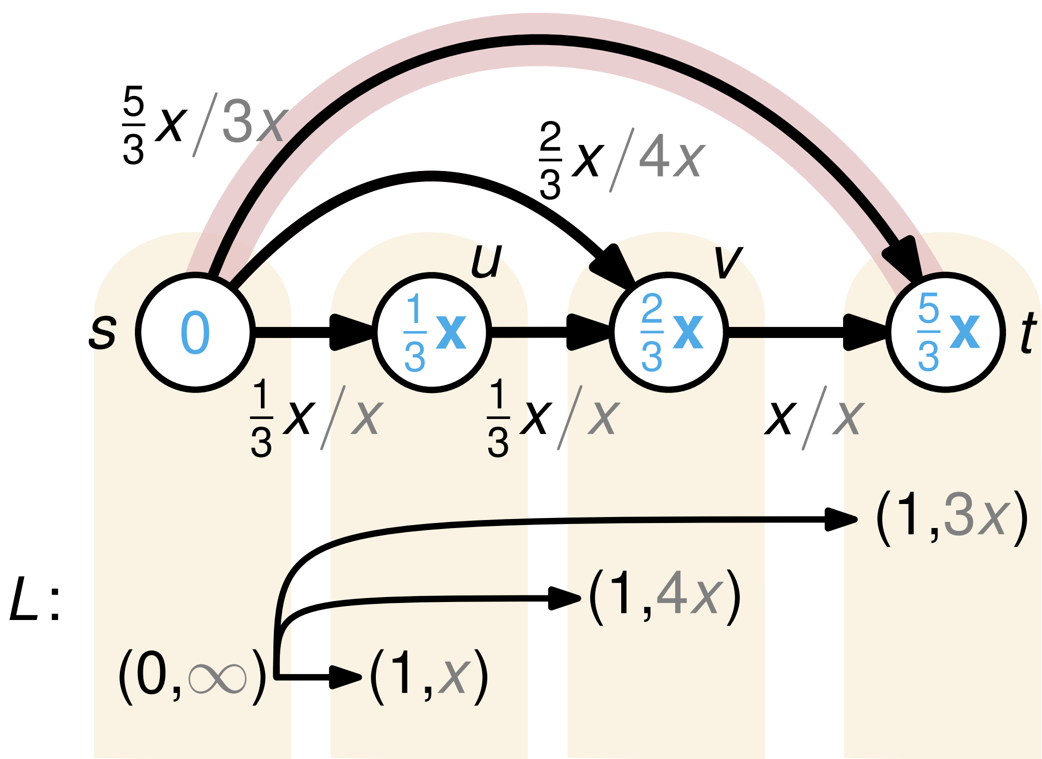
Institute of Theoretical Informatics
Algorithmics Group

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

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



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

Institute of Theoretical Informatics
Algorithmics Group

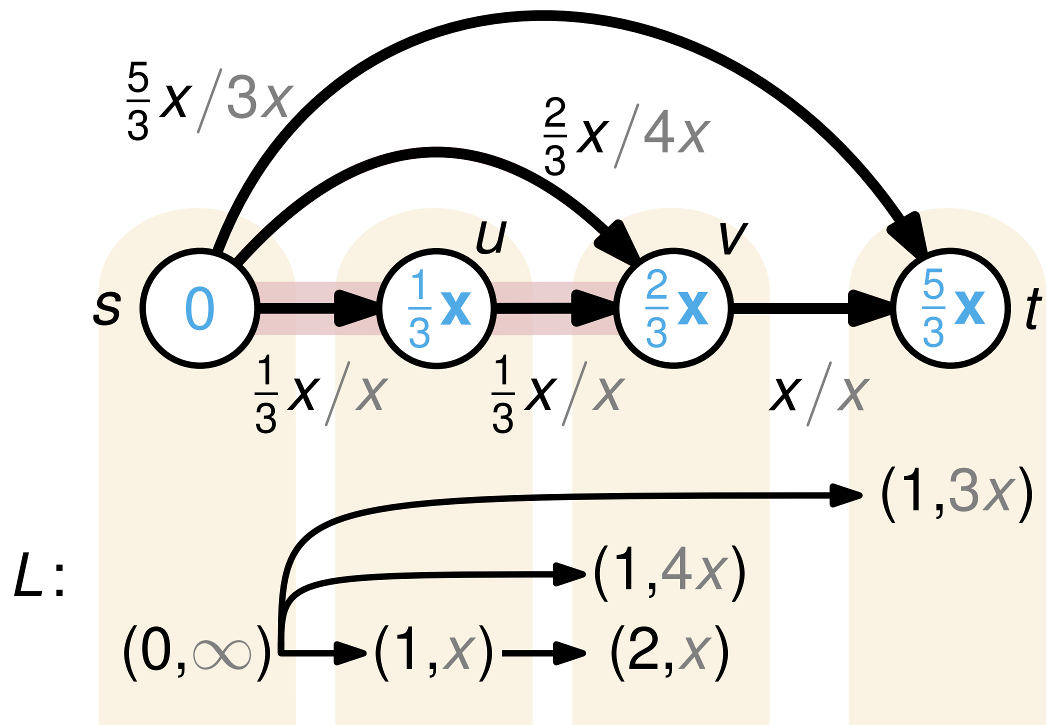
Computing DTP [Section 5; Grastien et al., 2018]

Karlsruhe Institute of Technology

Description:

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

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



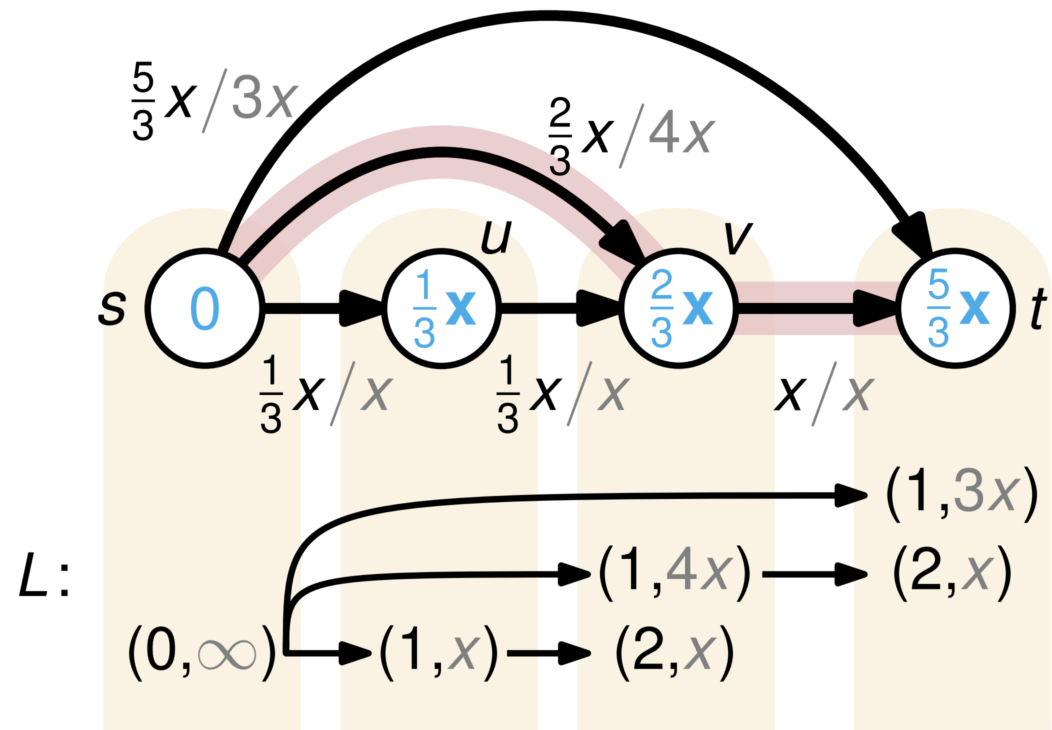
$\text{OPT}_{\text{MPF}} = \frac{8}{3}x$

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

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



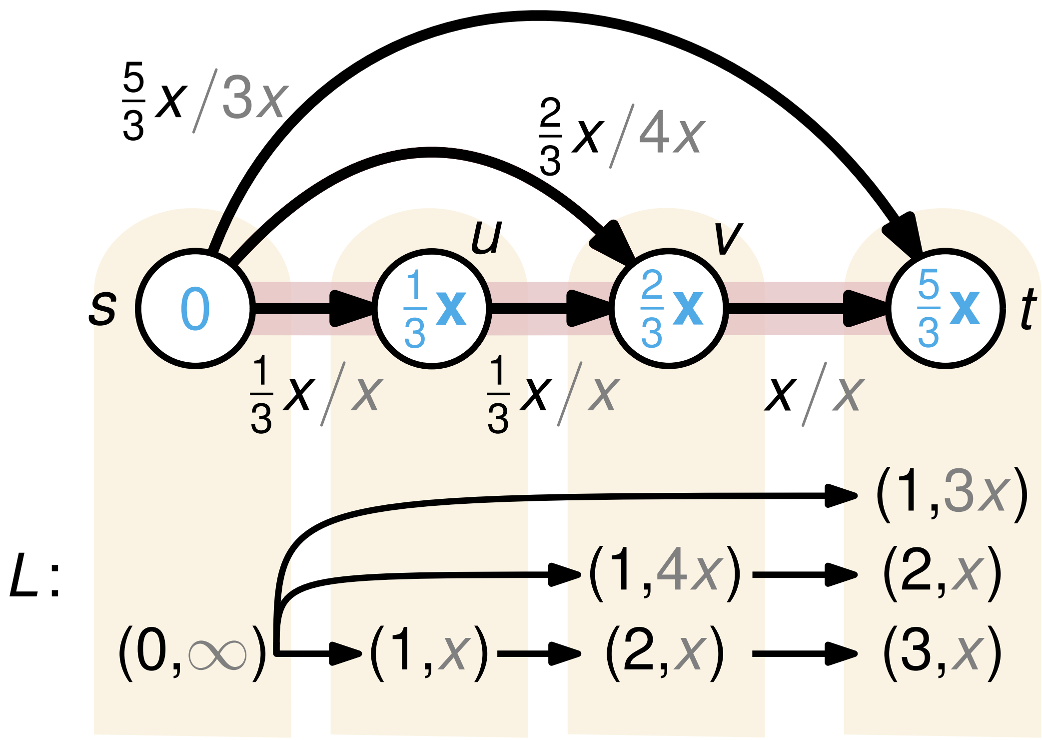
$\text{OPT}_{\text{MPF}} = \frac{8}{3}x$

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

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



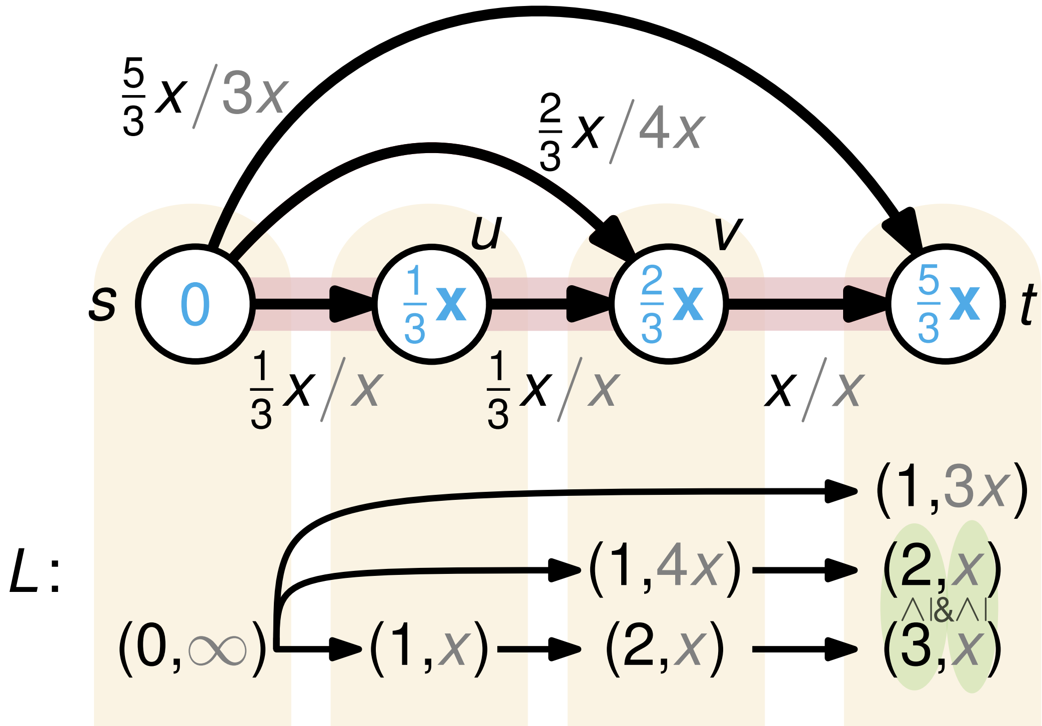
$\text{OPT}_{\text{MPF}} = \frac{8}{3}x$

Computing DTP [Section 5; Grastien et al., 2018]

Description:

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

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



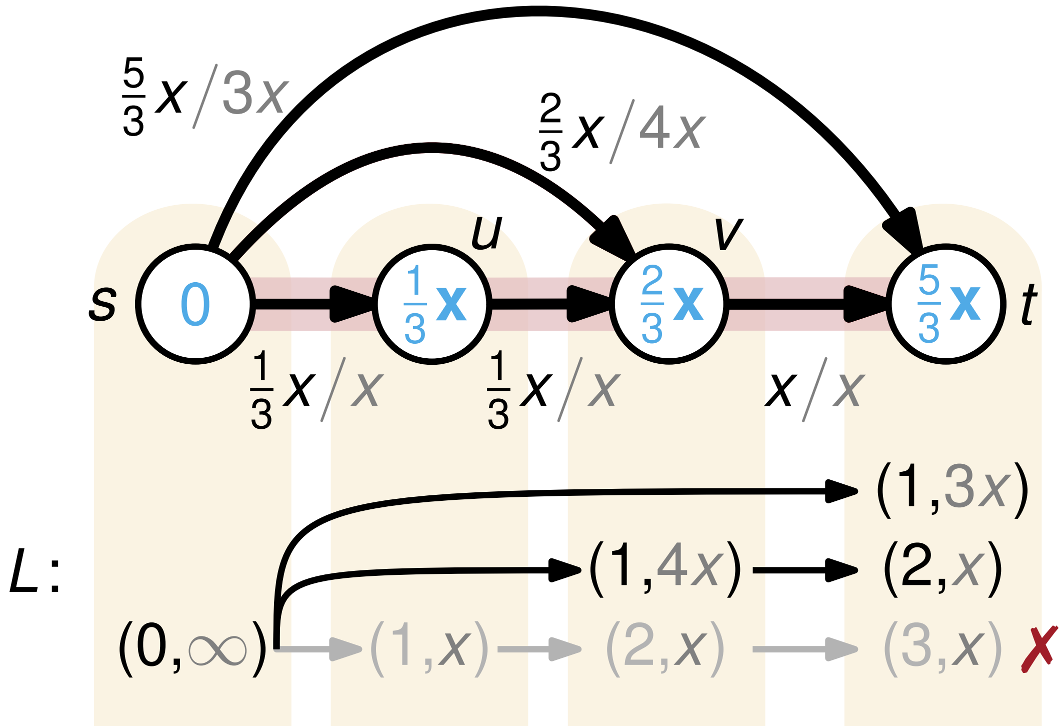
$\mathrm{OPT}_{\mathrm{MPF}} = \frac{8}{3}x$

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

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



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

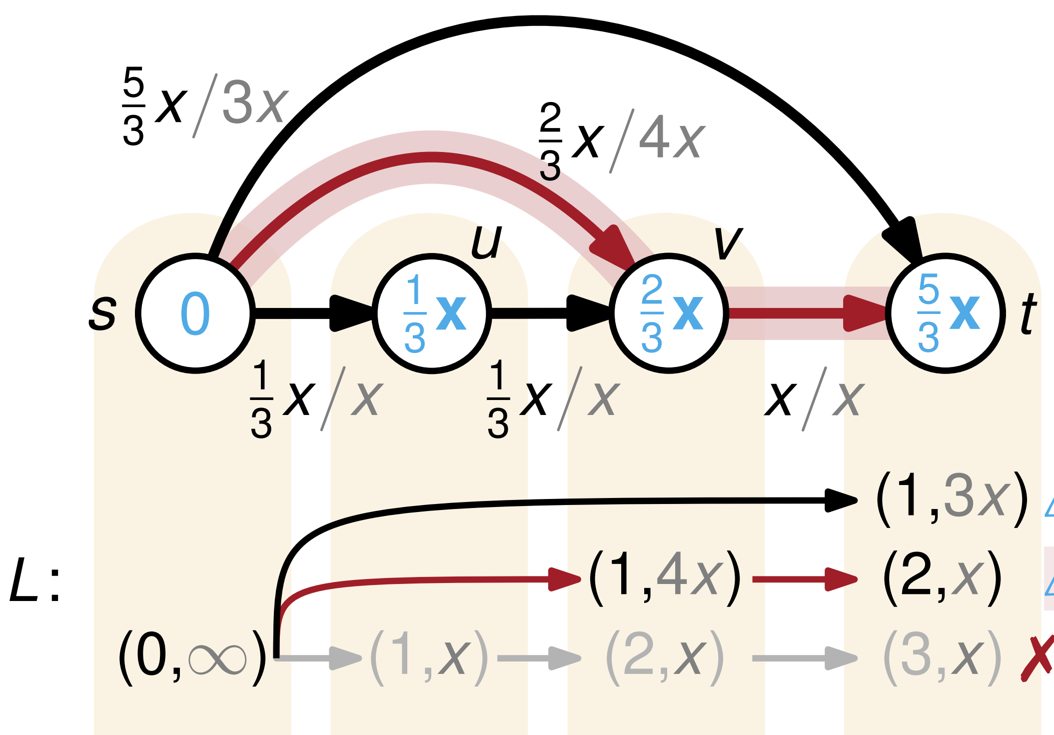
Institute of Theoretical Informatics
Algorithmics Group

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

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



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

$\Delta\theta(s,t) = 1 \cdot 3x = 3x$

$\vee|$

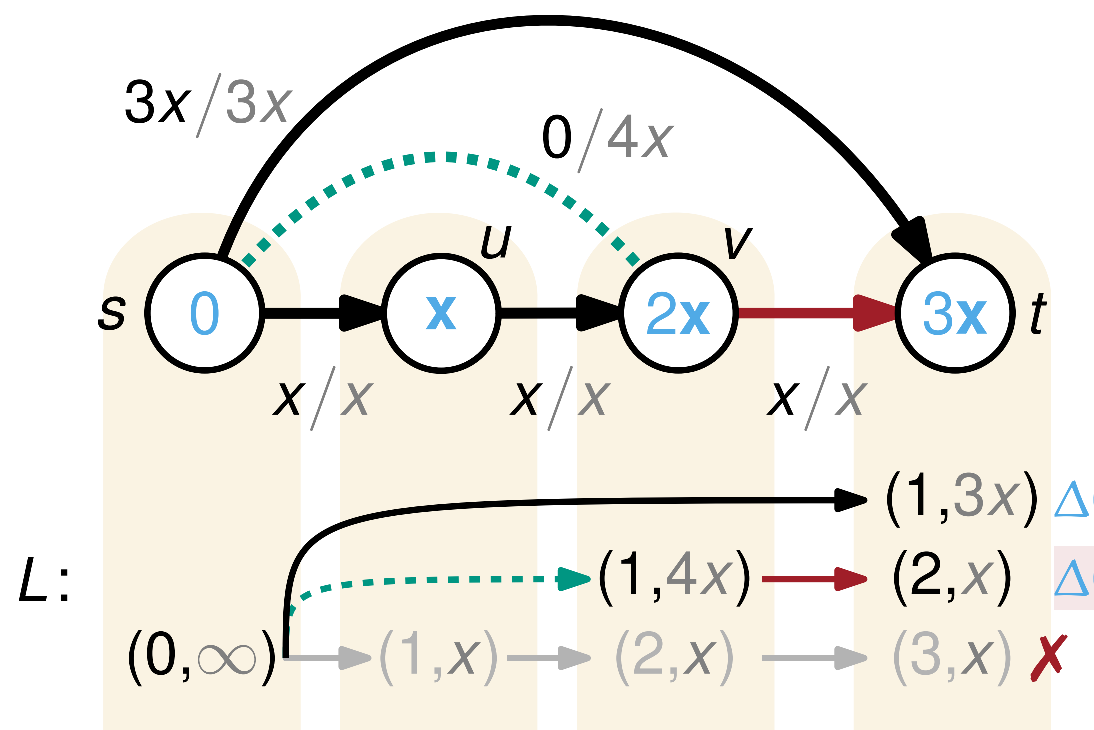
$\Delta\theta(s,v,t) = 2 \cdot x = 2x$

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

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



$\text{OPT}_{\text{MPF}} = 4x$
$= \text{OPT}_{\text{MTSF}}$

$\Delta\theta(s,t) = 1 \cdot 3x = 3x$

$\vee\!|$

$\Delta\theta(s,v,t) = 2 \cdot x = 2x$

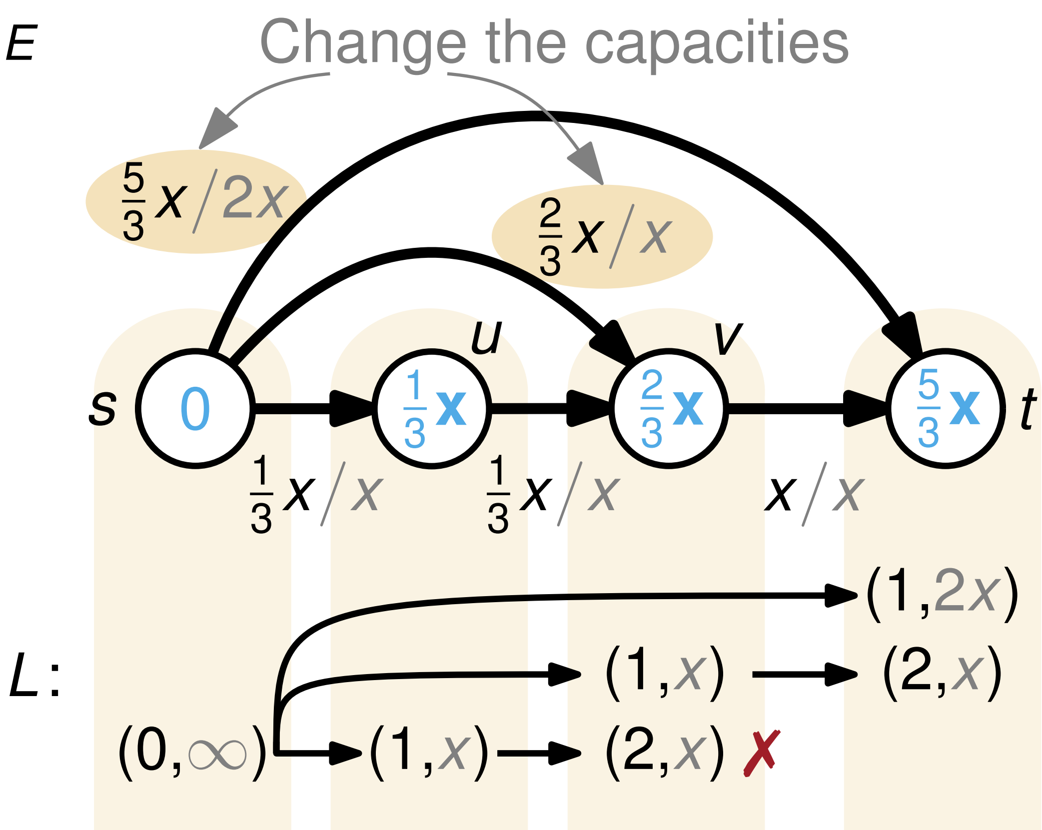
Institute of Theoretical Informatics
Algorithmics Group

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

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



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

Institute of Theoretical Informatics
Algorithmics Group

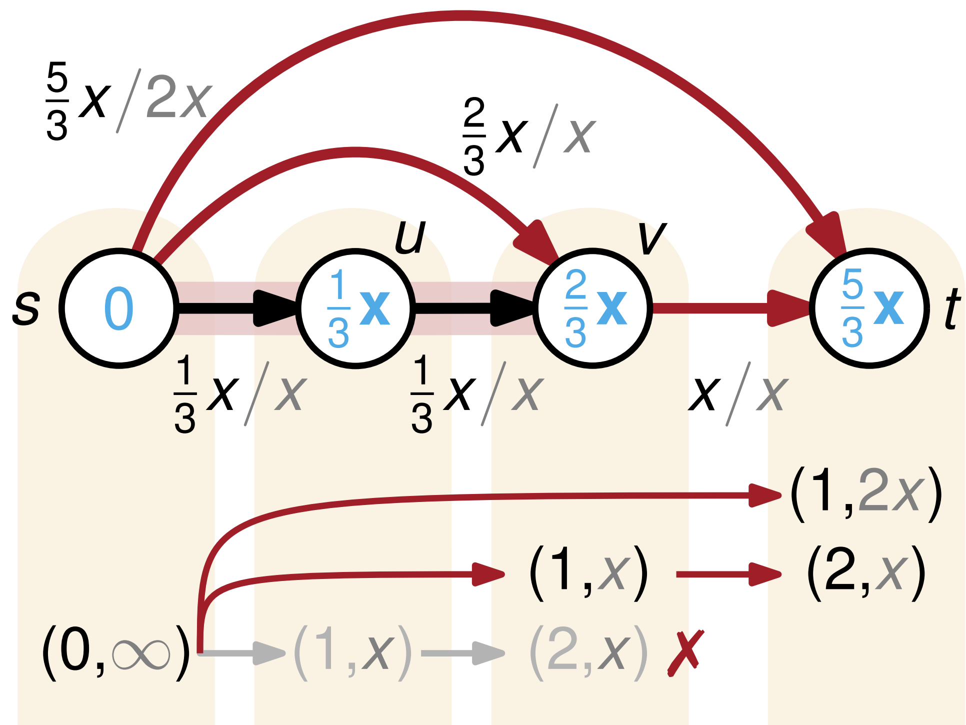
Computing DTP [Section 5; Grastien et al., 2018]

Description:

- Bicriterial Dijkstra with labels ($\|\pi\|_b$, $\underline{\mathrm{cap}}(\pi)$, 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



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

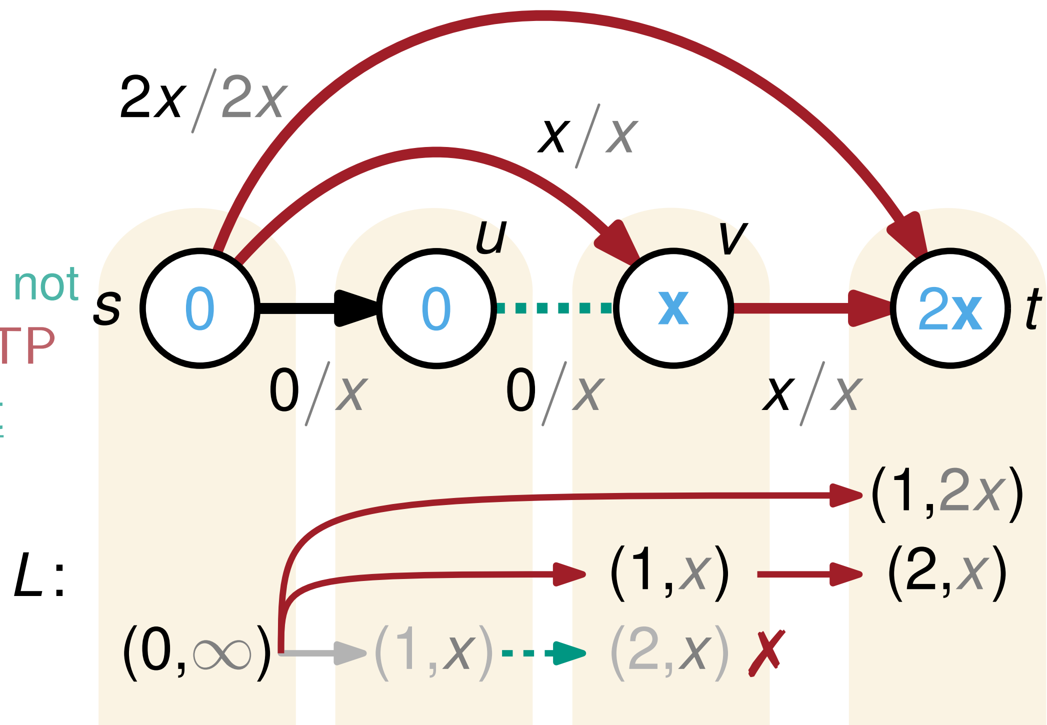
Computing DTP [Section 5; Grastien et al., 2018]

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



$OPT_{MPF} = 3x$
$= OPT_{MTSF}$

Institute of Theoretical Informatics
Algorithmics Group

Running Time Analysis for DTP

[page 345; Grastien et al., 2018]

Data: A network $\mathcal{N} = (G, V_G, V_C, \text{cap}, b)$.
Result: $\pi_{\text{DTP}}(s,t)$, $\Delta\theta_{\min}(s,t)$, and $D(v)$ with $v \in V$.

$D(u) := L(u) := \emptyset; \forall u \in V;$ ▷ Initialization
$Q := \emptyset;$
$L(s) := \{(0,\infty)\}$ ▷ Special label for source s
$Q.\texttt{insert}\big((0,\infty), s, \text{key}((0,\infty))\big);$
— $\mathcal{O}(|V|)$

while $Q \neq \emptyset$ **do** ▷ Visit all vertices — $|V|\cdot|E|$
  $(\ell, u, \text{key}) := Q.\texttt{deleteMin}();$ — $|V|\cdot|E|\cdot\mathcal{O}(\log|V|)$
  $D(u) := D(u) \cup \{\ell\};$
  for $\forall\{u,v\} \in E$ **do** ▷ Check adjacent vertices — $|E|^2$
    $\text{cap}(\pi(s,u,v)) := \min(\ell[1], \text{cap}(u,v));$ — $|E|^2\cdot\mathcal{O}(1)$
    $\ell_{\text{new}}(v) := \left(\ell[0] + \frac{1}{b(u,v)}, \text{cap}(\pi(s,u,v))\right);$ — $|E|^2\cdot\mathcal{O}(1)$
    If $\texttt{isReachable}(V \setminus \{v\}, \ell, s)$ **then** — $|E|^2\cdot\mathcal{O}\left(2^{|V|}|V|\cdot|E|\right)$
      If $\ell_{\text{new}}(v) \in L(v)$ **then**
        $\text{parent}(\ell_{\text{new}}(v)) := \text{parent}(\ell_{\text{new}}(v)) \cup \{\ell\};$ — $|E|^2\cdot\mathcal{O}(1)$
      else if not $L(v)$ **dominates** $\ell_{\text{new}}(v)$ **then**
        $L(v).\texttt{deleteDominatedLabels}(\ell_{\text{new}}(v));$ — $|E|^2\cdot\mathcal{O}(|E|)$
        $Q.\texttt{deleteDominatedLabels}(\ell_{\text{new}}(v), v);$ — $|E|^2\cdot\mathcal{O}(|E|)$
        $L(v).\texttt{insert}(\ell_{\text{new}}(v));$ — $|E|^2\cdot\mathcal{O}(1)$
        $Q.\texttt{insert}(\ell_{\text{new}}(v), v, \text{key}(\ell_{\text{new}}(v)));$ — $|E|^2\cdot\mathcal{O}(1)$
        $\text{parent}(\ell_{\text{new}}(v)) := \{\ell\};$ — $|E|^2\cdot\mathcal{O}(1)$
      end
    end
  end
end

return $\begin{pmatrix} \pi_{\text{DTP}}(s,t) := \texttt{getPaths}(s,t), \\ \Delta\theta_{\min}(s,t) := \min_{\ell\in D(t)}\{\ell[0]\cdot\ell[1]\}, \\ D(\cdot) \end{pmatrix}$ ▷ Build paths from parent

Each vertex has at most $|E|$ labels

Königsberg Bridge Problem

$\sum_{v\in V}\deg(v) = 2|E|$

iterate over all labels in the bag, $|E|$ tests per bag

Fibonacci-heap $\mathcal{Q}$

`insert`	$\mathcal{O}(1)$	
`decreaseKey`	$\mathcal{O}(1)$	*amortized*
`delMin`	$\mathcal{O}(\log|V|)$	*amortized*

Institute of Theoretical Informatics
Algorithms Group

Running Time Analysis for DTP

[page 345; Grastien et al., 2018]

Karlsruhe Institute of Technology

Data: A network $\mathcal{N} = (G, V_G, V_C, \text{cap}, b)$.
Result: $\pi_{\text{DTP}}(s,t)$, $\Delta\theta_{\min}(s,t)$, and $D(v)$ with $v \in V$.

```
D(u) := L(u) := ∅; ∀u ∈ V;                          ▷ Initialization          }
Q := ∅;                                                                         }  O(|V|)
L(s) := {(0, ∞)}                                    ▷ Special label for source s }
Q.insert((0, ∞), s, key((0, ∞)));                                               }
while Q ≠ ∅ do                                      ▷ Visit all vertices        |V|·|E|
    (ℓ, u, key) := Q.deleteMin();                                              |V|·|E|·O(log|V|)
    D(u) := D(u) ∪ {ℓ};
    for ∀{u, v} ∈ E do                              ▷ Check adjacent vertices   |E|^2
            parent(ℓ_new(v)) := parent(ℓ_new(v)) ∪ {ℓ};                         |E|^2·O(1)
        else if not L(v) dominates ℓ_new(v) then
            L(v).deleteDominatedLabels(ℓ_new(v));                               |E|^2·O(|E|)
            Q.deleteDominatedLabels(ℓ_new(v), v);                               |E|^2·O(|E|)
            L(v).insert(ℓ_new(v));                                              |E|^2·O(1)
            Q.insert(ℓ_new(v), v, key(ℓ_new(v)));                               |E|^2·O(1)
            parent(ℓ_new(v)) := {ℓ};                                            |E|^2·O(1)
        end
      end
    end
end
return ( π_DTP(s,t) := getPaths(s,t),          ▷ Build paths from parent
         Δθ_min(s,t) := min_{ℓ∈D(t)} {ℓ[0] · ℓ[1]},
         D(·) )
```

Each vertex has at m…

Königsberg Bridge P…

$\sum_{v \in V} \deg(v) =$

iterate over all labels in the bag, $|E|$ tests per bag

Fibonacci-heap Q		
insert	$\mathcal{O}(1)$	
decreaseKey	$\mathcal{O}(1)$	*amortized*
delMin	$\mathcal{O}(\log\|V\|)$	*amortized*

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

Institute of Theoretical Informatics
Algorithmics Group

Running Time Analysis for DTP

[page 345; Grastien et al., 2018]

Data: A network $\mathcal{N} = (G, V_G, V_C, \text{cap}, b)$.
Result: $\pi_{\text{DTP}}(s,t)$, $\Delta\theta_{\min}(s,t)$, and $D(v)$ with $v \in V$.

```
D(u) := L(u) := ∅; ∀u ∈ V;                               ▷ Initialization      }
Q := ∅;                                                                         }  O(|V|)
L(s) := {(0, ∞)}                                ▷ Special label for source s   }
Q.insert((0, ∞), s, key((0, ∞)));                                               }
while Q ≠ ∅ do                                         ▷ Visit all vertices      |V|·|E|
    (ℓ, u, key) := Q.deleteMin();                                                |V|·|E|·O(log|V|)
    D(u) := D(u) ∪ {ℓ};
    for ∀{u, v} ∈ E do                              ▷ Check adjacent vertices    |E|^2
        cap(π(s, u, v)) := min(ℓ[1], cap(u, v));                                 |E|^2·O(1)
        ...
        if ℓ_new(v) ∈ L(v) then
            parent(ℓ_new(v)) := parent(ℓ_new(v)) ∪ {ℓ};                          |E|^2·O(1)
        else if not L(v) dominates ℓ_new(v) then
            L(v).deleteDominatedLabels(ℓ_new(v));                                |E|^2·O(|E|)
            Q.deleteDominatedLabels(ℓ_new(v), v);                                |E|^2·O(|E|)
            L(v).insert(ℓ_new(v));                                               |E|^2·O(1)
            Q.insert(ℓ_new(v), v, key(ℓ_new(v)));                                |E|^2·O(1)
            parent(ℓ_new(v)) := {ℓ};                                             |E|^2·O(1)
        end
      end
    end
end
return ( π_DTP(s, t) := getPaths(s, t),       ▷ Build paths from parent
         Δθ_min(s, t) := min_{ℓ∈D(t)} {ℓ[0] · ℓ[1]},
         D(·) )
```

Each vertex has at m...

Königsberg Bridge P...

$\sum_{v \in V} \deg(v) = \ldots$

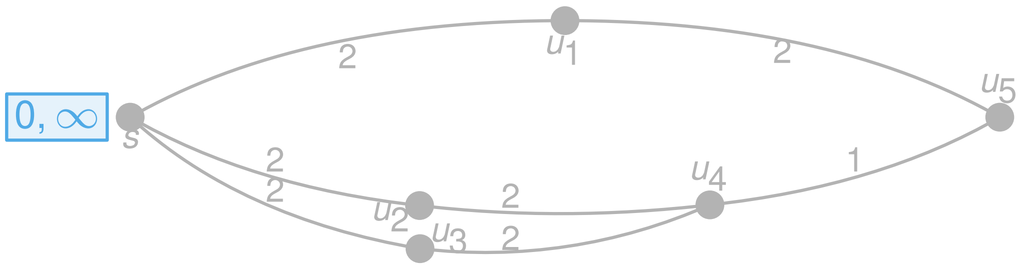
iterate over all labels in the bag, $|E|$ tests per bag

Fibonacci-heap Q		
insert	$\mathcal{O}(1)$	
decreaseKey	$\mathcal{O}(1)$	amortized
delMin	$\mathcal{O}(\log \lvert V \rvert)$	amortized

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

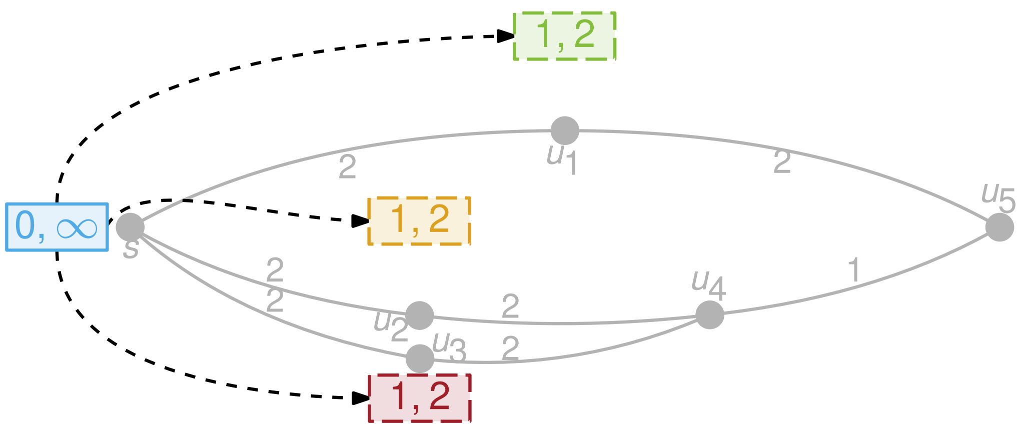
Institute of Theoretical Informatics
Algorithmics Group

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

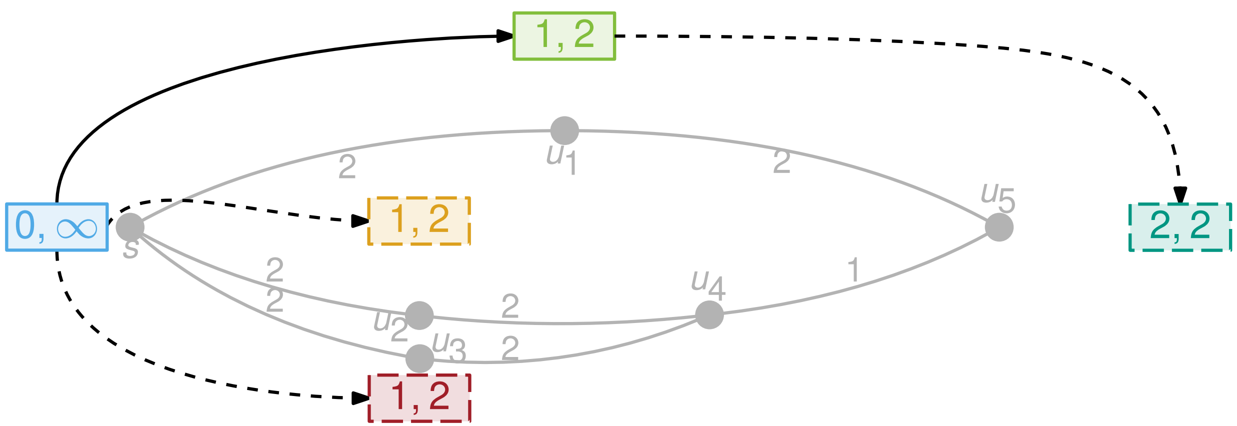


Institute of Theoretical Informatics
Algorithmics Group

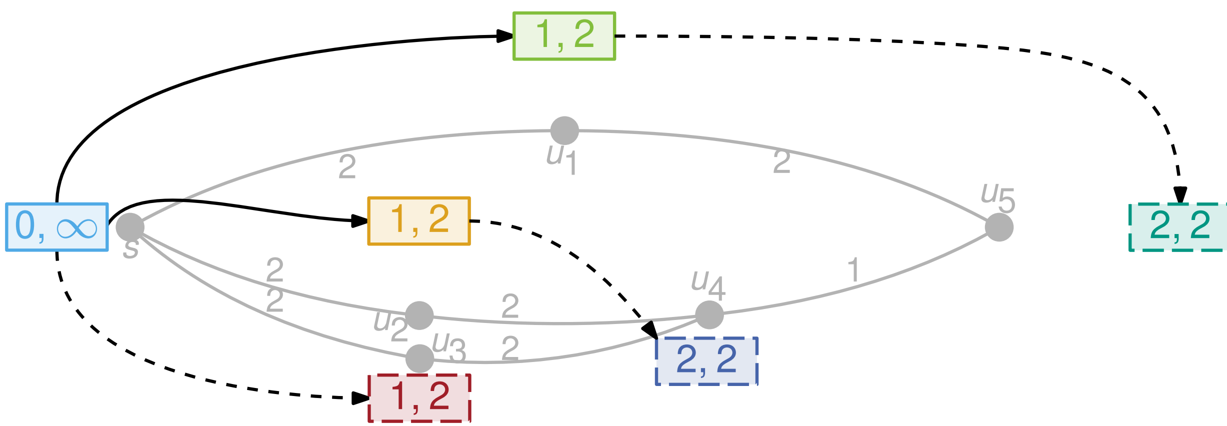
isReachable [page 354; Grastien et al., 2018]



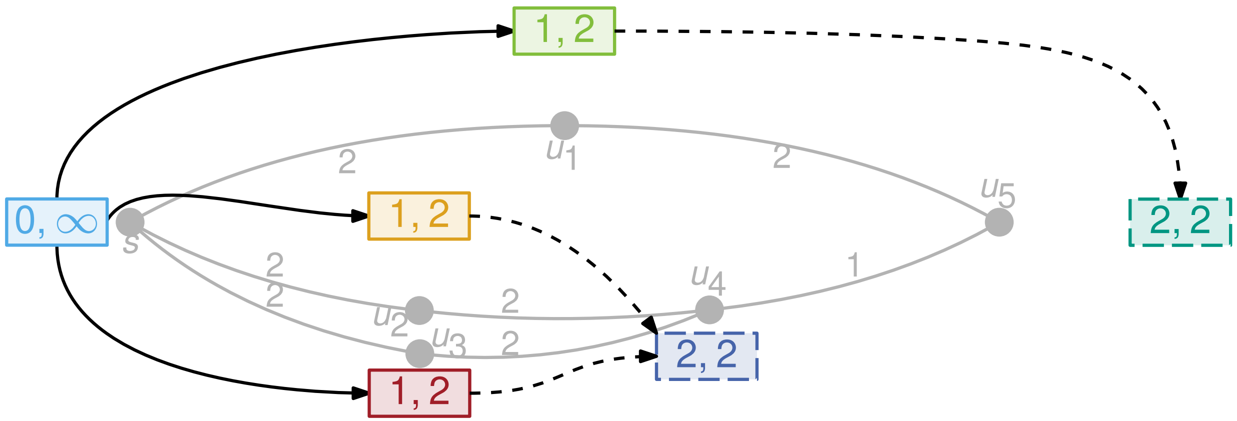
isReachable [page 354; Grastien et al., 2018]



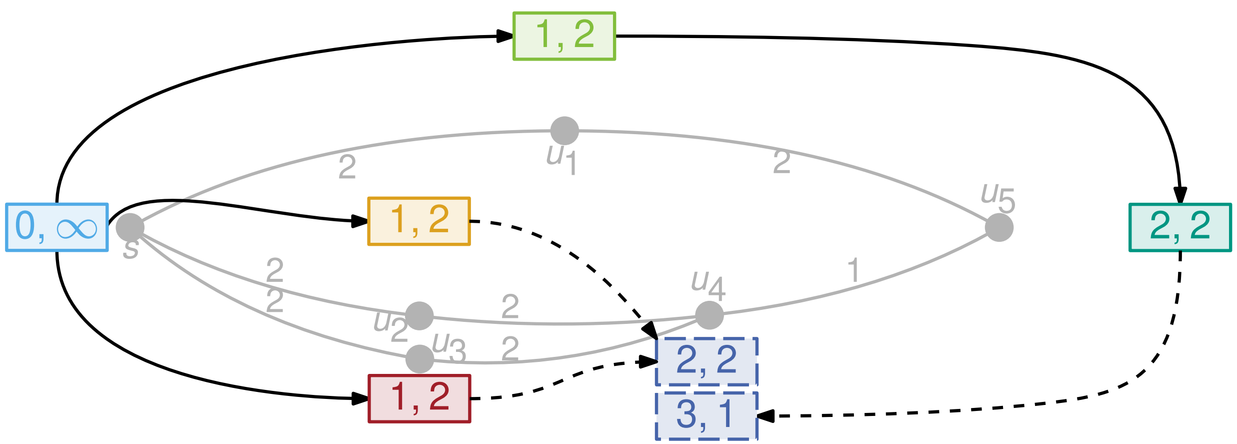
isReachable [page 354; Grastien et al., 2018]



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

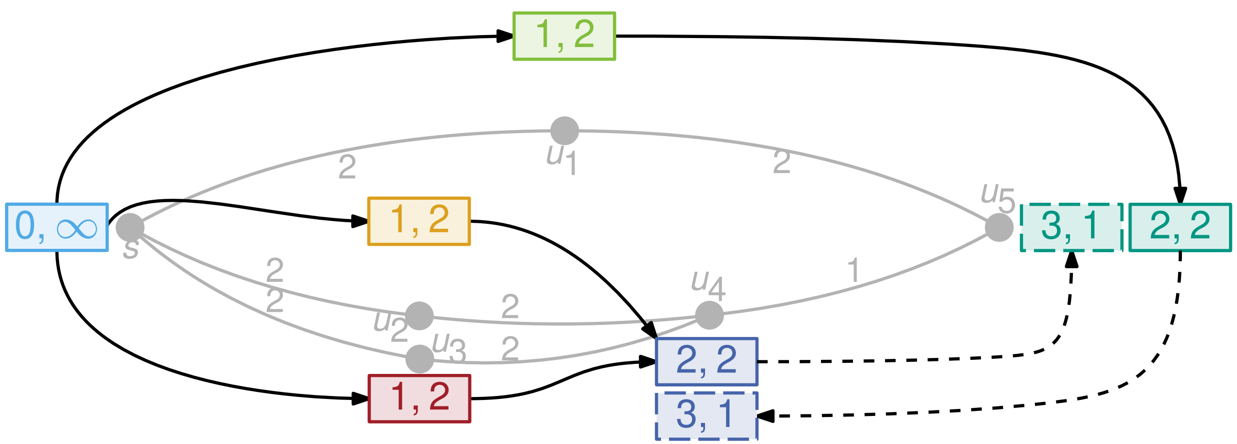


isReachable [page 354; Grastien et al., 2018]

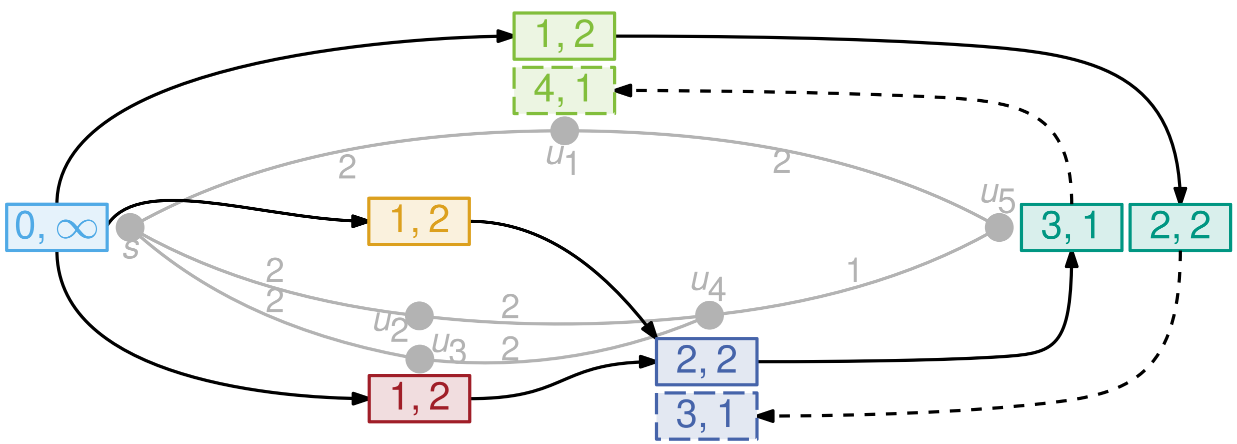


Institute of Theoretical Informatics
Algorithmics Group

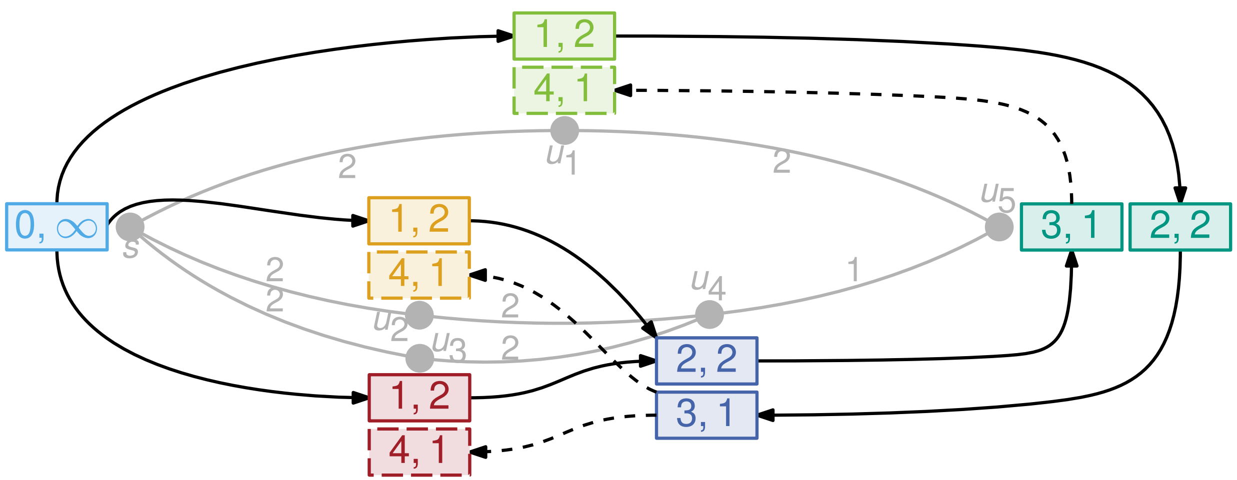
isReachable [page 354; Grastien et al., 2018]



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

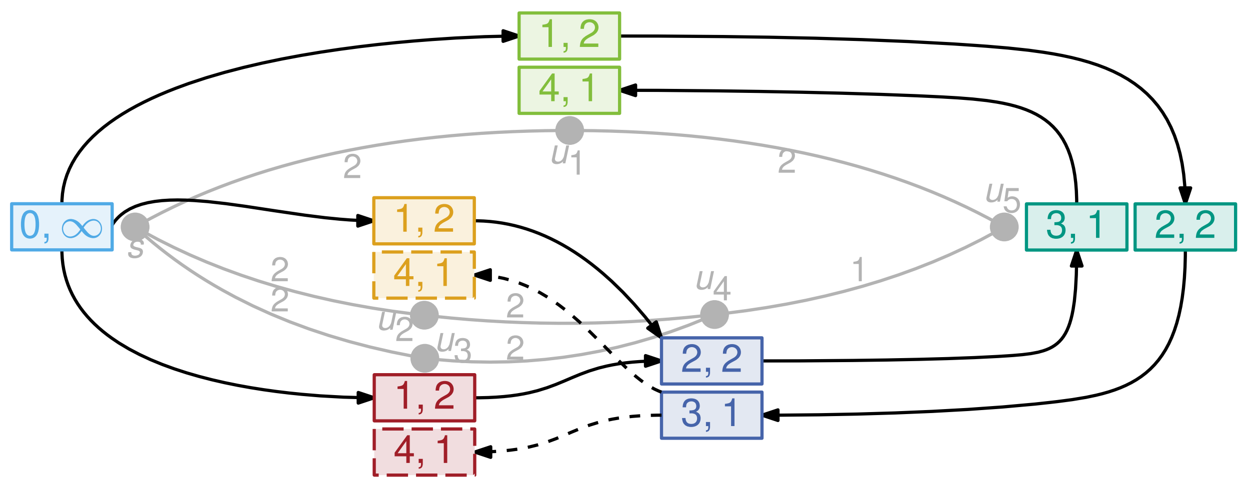


isReachable [page 354; Grastien et al., 2018]



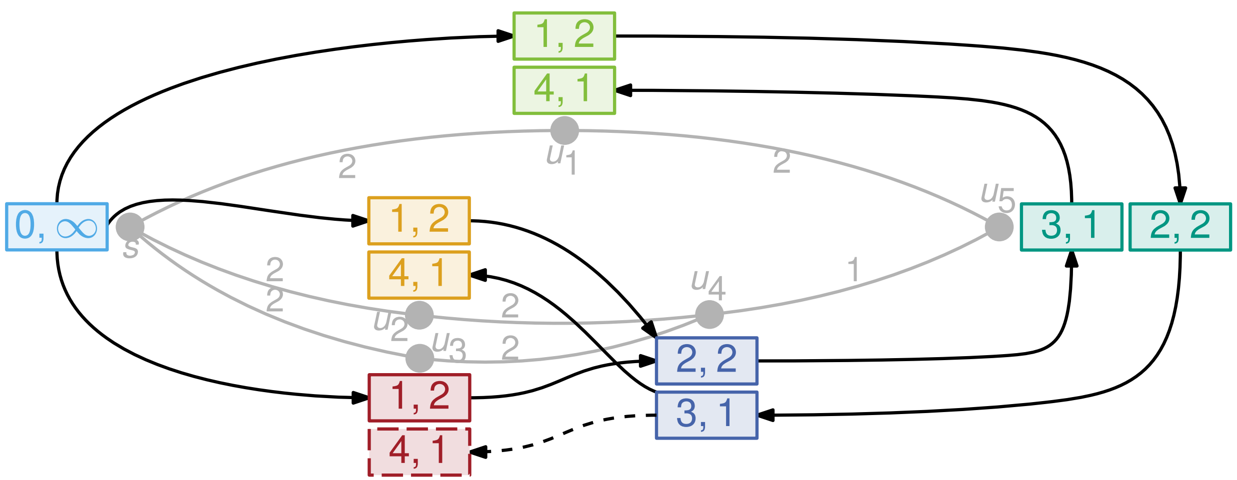
Institute of Theoretical Informatics
Algorithmics Group

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

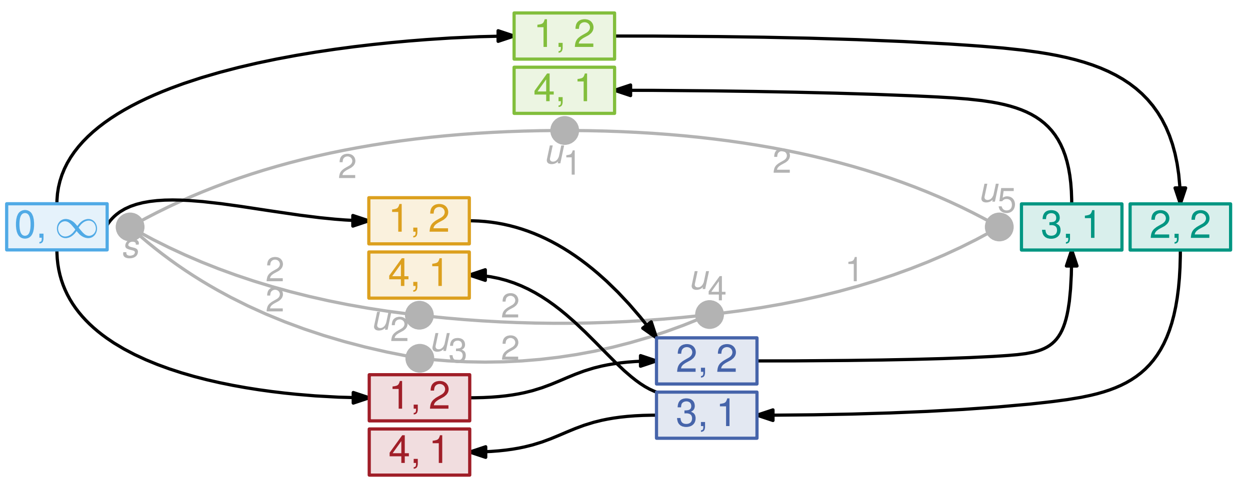


Institute of Theoretical Informatics
Algorithmics Group

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

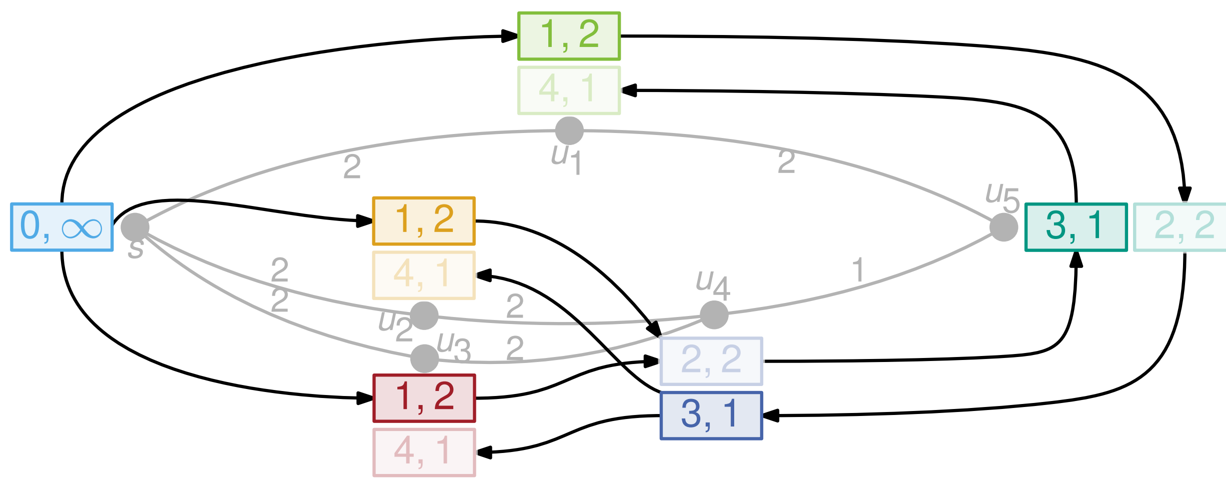


`isReachable` [page 354; Grastien et al., 2018]



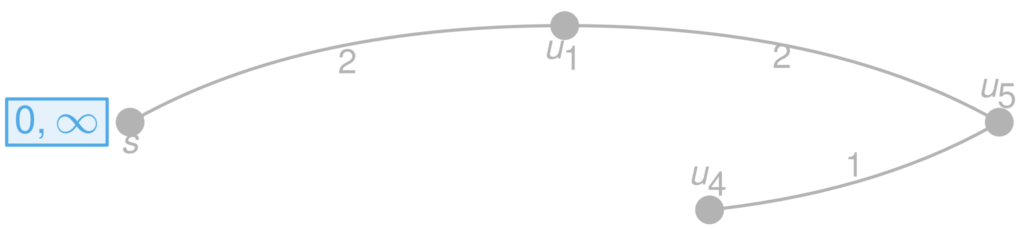
Institute of Theoretical Informatics
Algorithmics Group

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



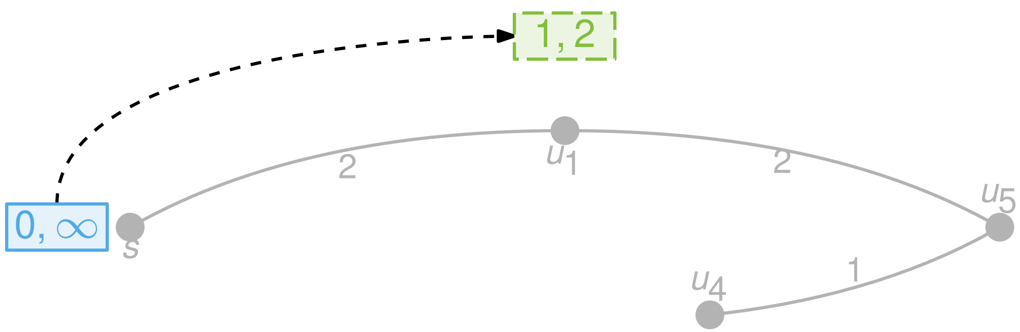
isReachable [page 354; Grastien et al., 2018]

- Does it always work like this?



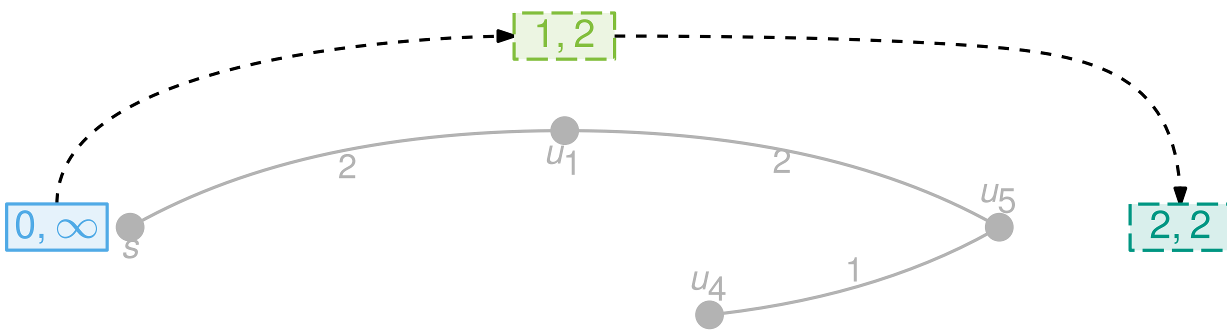
`isReachable` [page 354; Grastien et al., 2018]

- Does it always work like this?



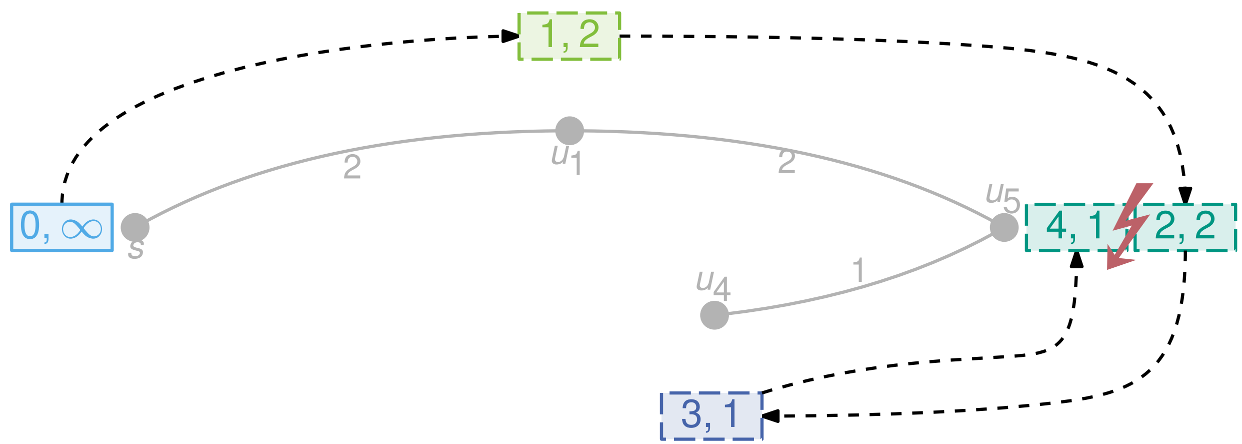
isReachable [page 354; Grastien et al., 2018]

- Does it always work like this?



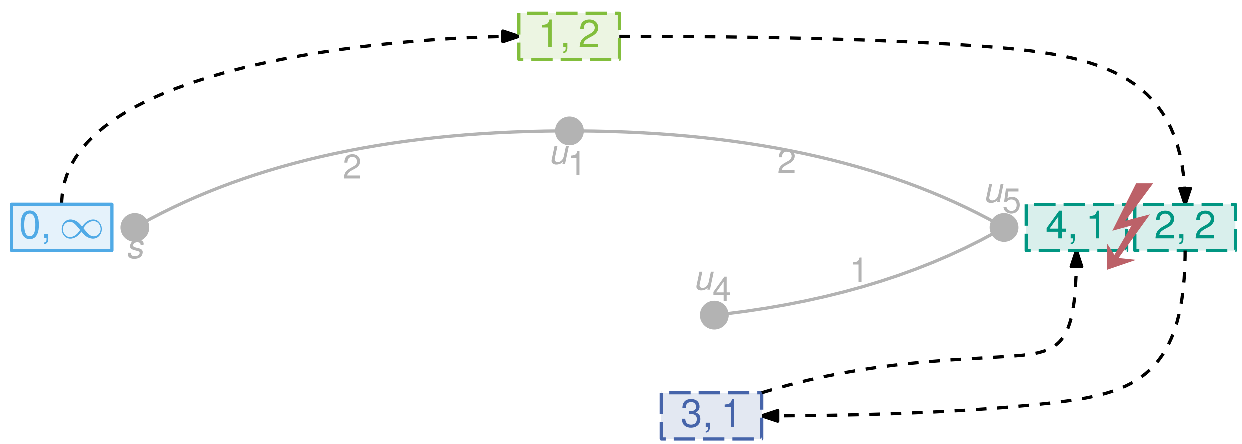
isReachable [page 354; Grastien et al., 2018]

- Does it always work like this?



Institute of Theoretical Informatics
Algorithmics Group

- 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-RP)

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

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

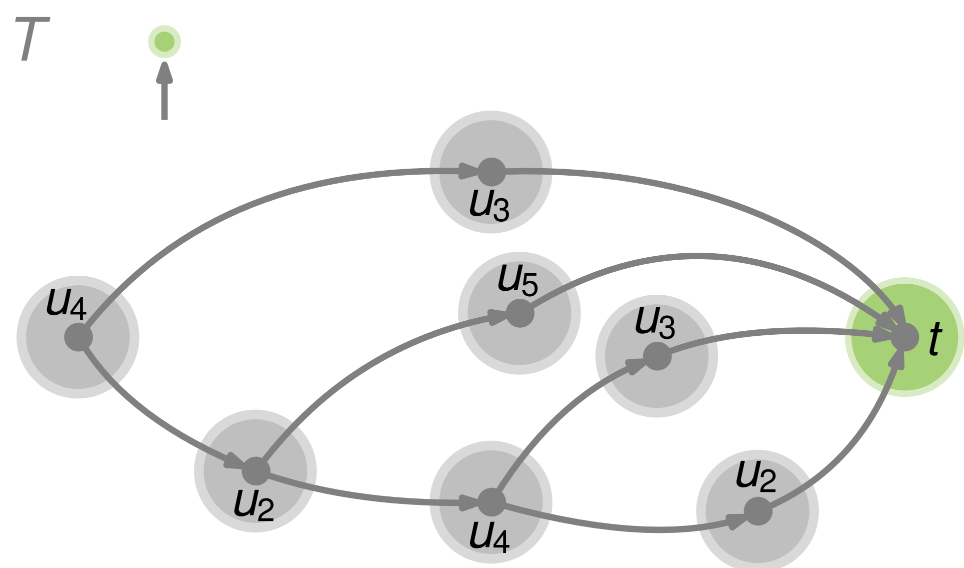
But: n colors $\Rightarrow$ not a polynomial time algorithm(?)

Institute of Theoretical Informatics
Algorithmics Group

Rainbow Paths – FPT Algorithm

[Theorem 11; Uchizawa et al., 2013]

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



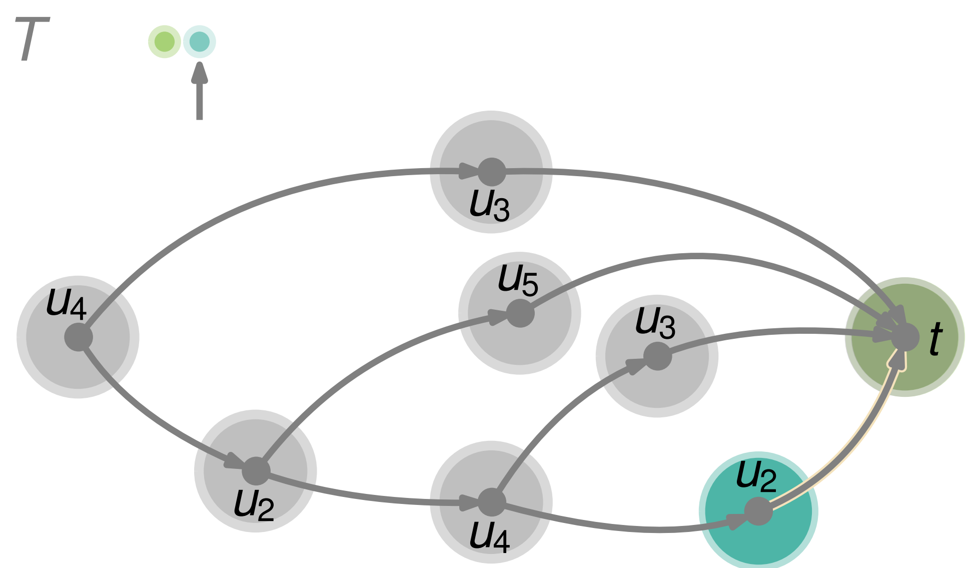
But: n colors $\Rightarrow$ not a polynomial time algorithm(?)

Institute of Theoretical Informatics
Algorithmics Group

Rainbow Paths – FPT Algorithm

[Theorem 11; Uchizawa et al., 2013]

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

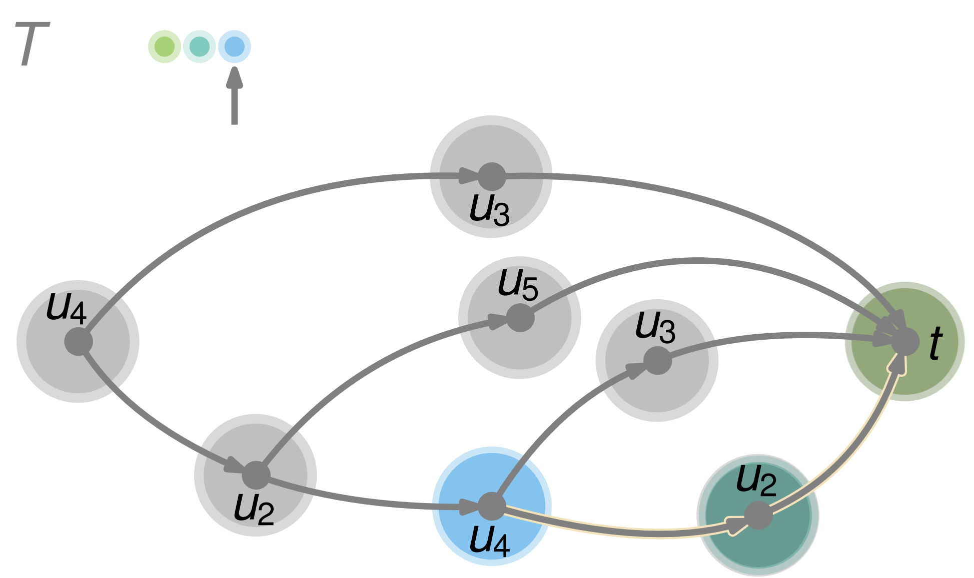


But: n colors $\Rightarrow$ not a polynomial time algorithm(?)

Rainbow Paths – FPT Algorithm

[Theorem 11; Uchizawa et al., 2013]

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

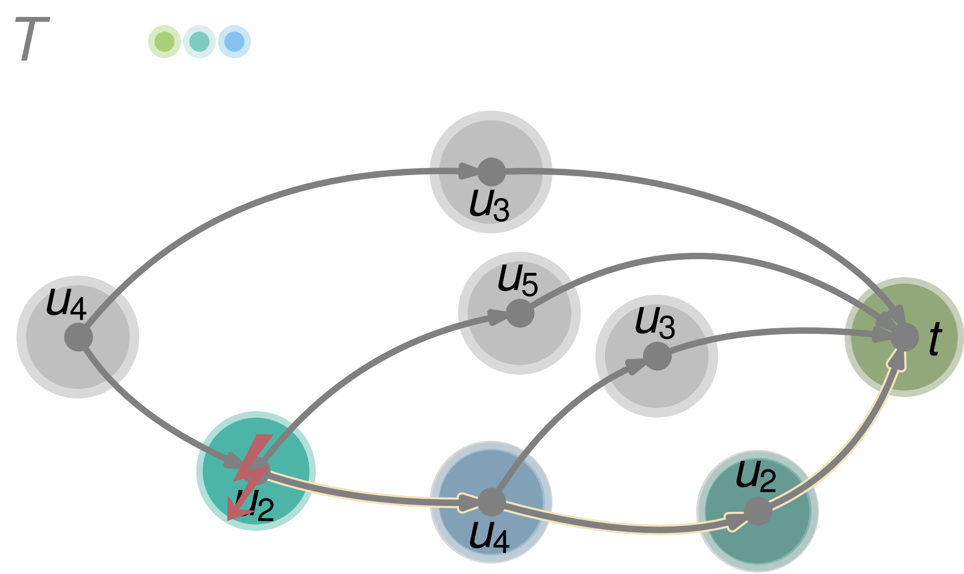


But: n colors $\Rightarrow$ not a polynomial time algorithm(?)

Rainbow Paths – FPT Algorithm

[Theorem 11; Uchizawa et al., 2013]

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

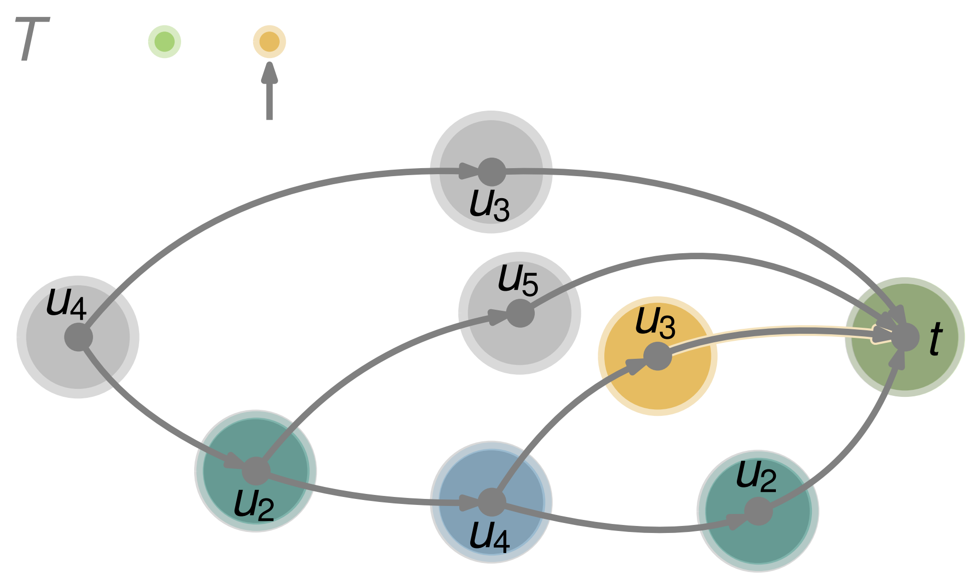


But: n colors $\Rightarrow$ not a polynomial time algorithm(?)

Rainbow Paths – FPT Algorithm

[Theorem 11; Uchizawa et al., 2013]

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

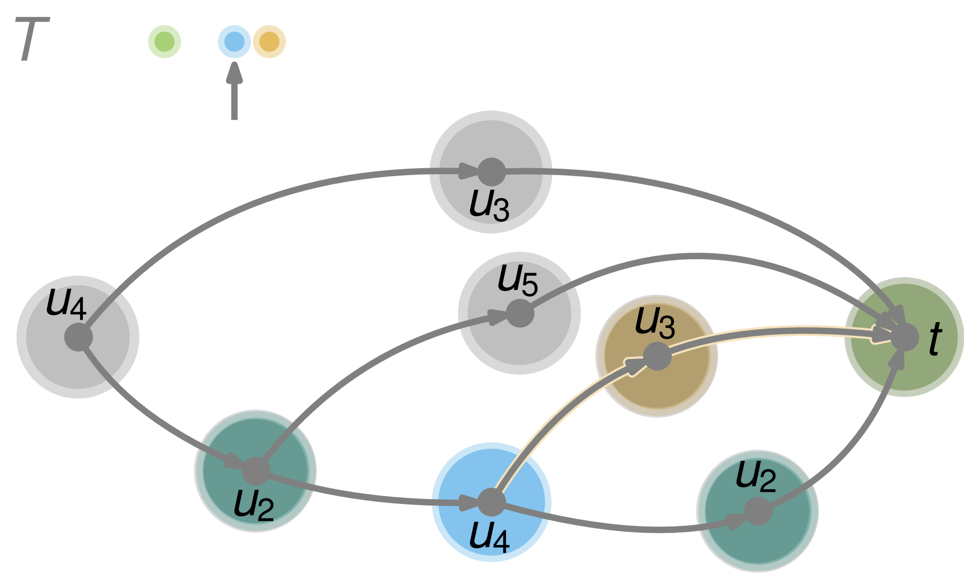


But: n colors $\Rightarrow$ not a polynomial time algorithm(?)

Rainbow Paths – FPT Algorithm

[Theorem 11; Uchizawa et al., 2013]

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

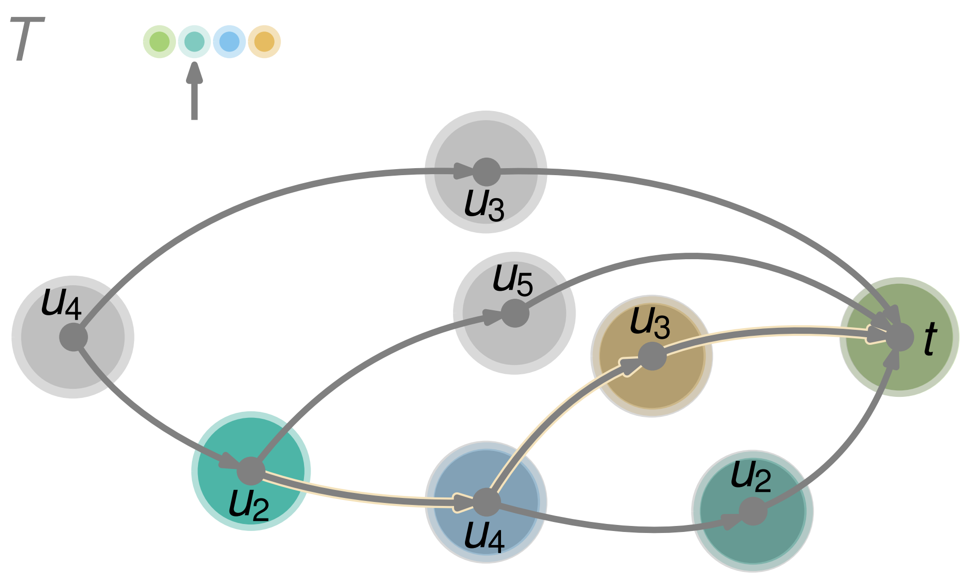


But: n colors $\Rightarrow$ not a polynomial time algorithm(?)

Rainbow Paths – FPT Algorithm

[Theorem 11; Uchizawa et al., 2013]

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

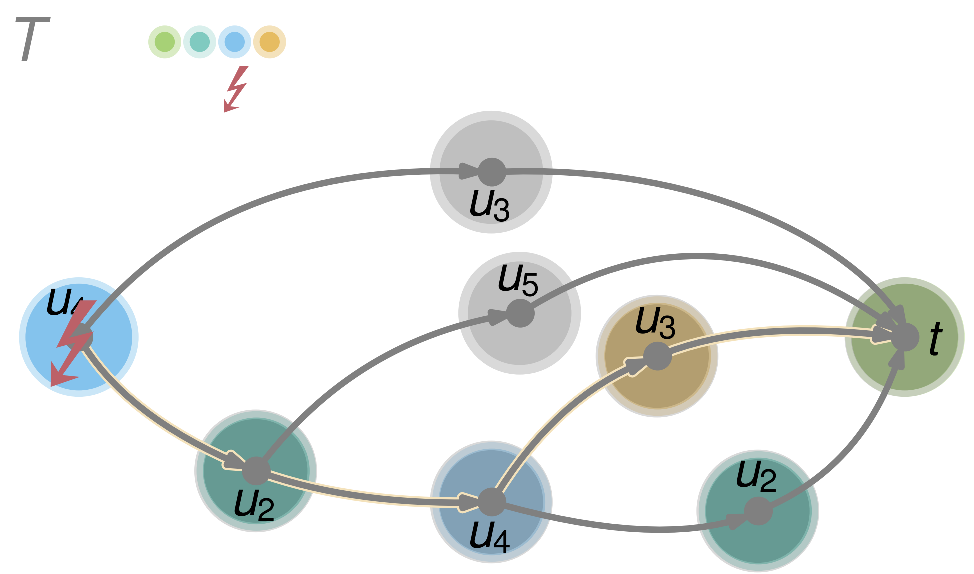


But: n colors $\Rightarrow$ not a polynomial time algorithm(?)

Rainbow Paths – FPT Algorithm

[Theorem 11; Uchizawa et al., 2013]

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



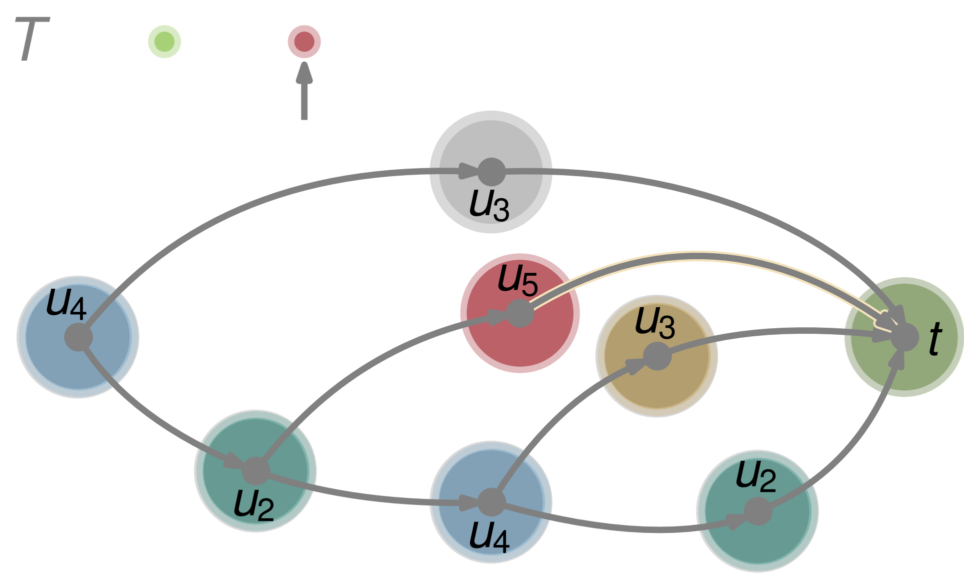
But: n colors $\Rightarrow$ not a polynomial time algorithm(?)

Institute of Theoretical Informatics
Algorithmics Group

Rainbow Paths – FPT Algorithm

[Theorem 11; Uchizawa et al., 2013]

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



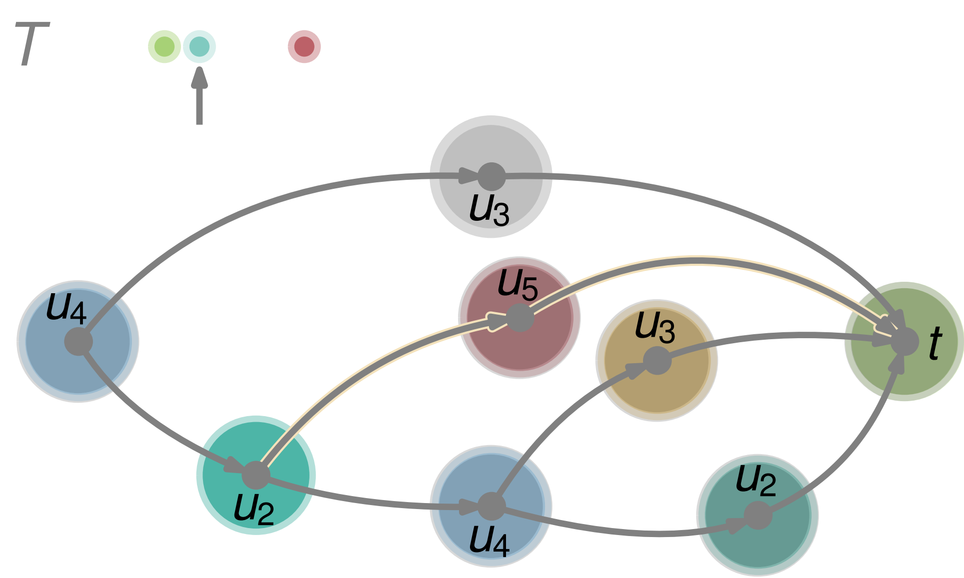
But: n colors $\Rightarrow$ not a polynomial time algorithm(?)

Institute of Theoretical Informatics
Algorithmics Group

Rainbow Paths – FPT Algorithm

[Theorem 11; Uchizawa et al., 2013]

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

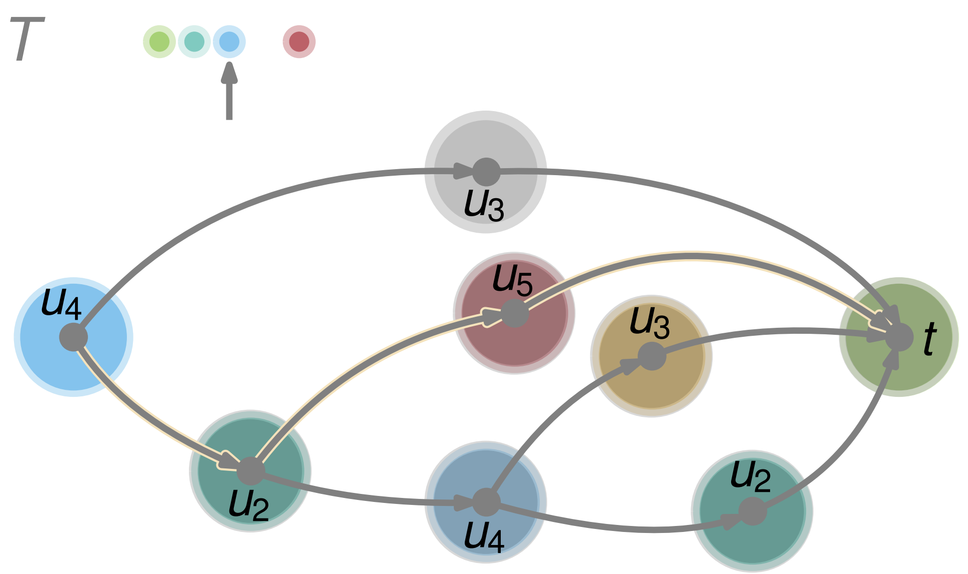


But: n colors $\Rightarrow$ not a polynomial time algorithm(?)

Rainbow Paths – FPT Algorithm

[Theorem 11; Uchizawa et al., 2013]

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

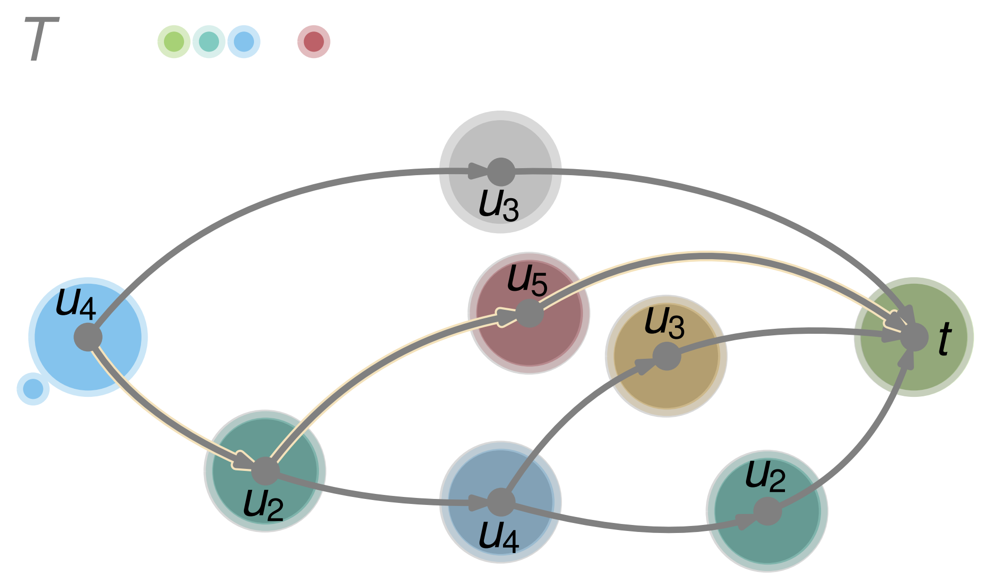


But: n colors $\Rightarrow$ not a polynomial time algorithm(?)

Rainbow Paths – FPT Algorithm

[Theorem 11; Uchizawa et al., 2013]

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

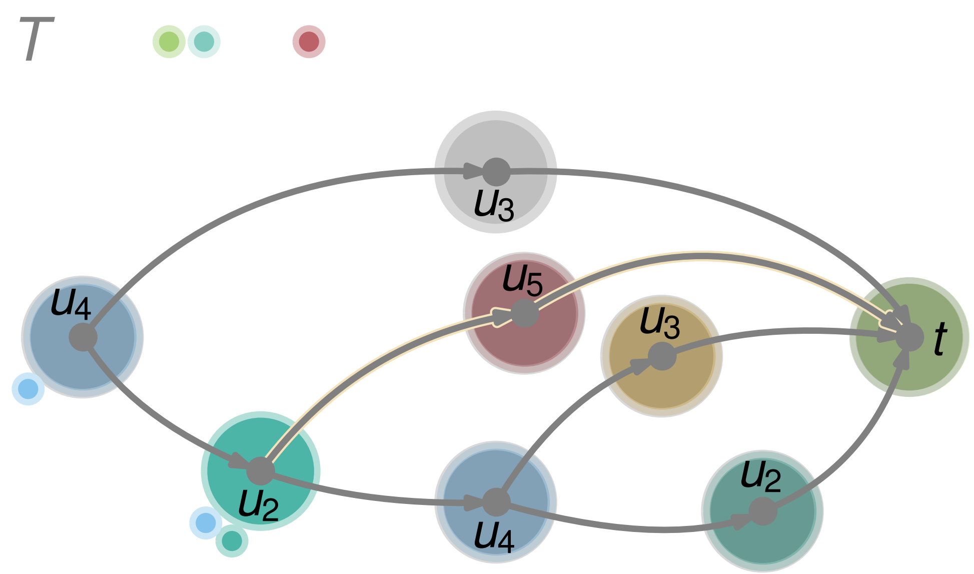


But: n colors $\Rightarrow$ not a polynomial time algorithm(?)

Rainbow Paths – FPT Algorithm

[Theorem 11; Uchizawa et al., 2013]

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

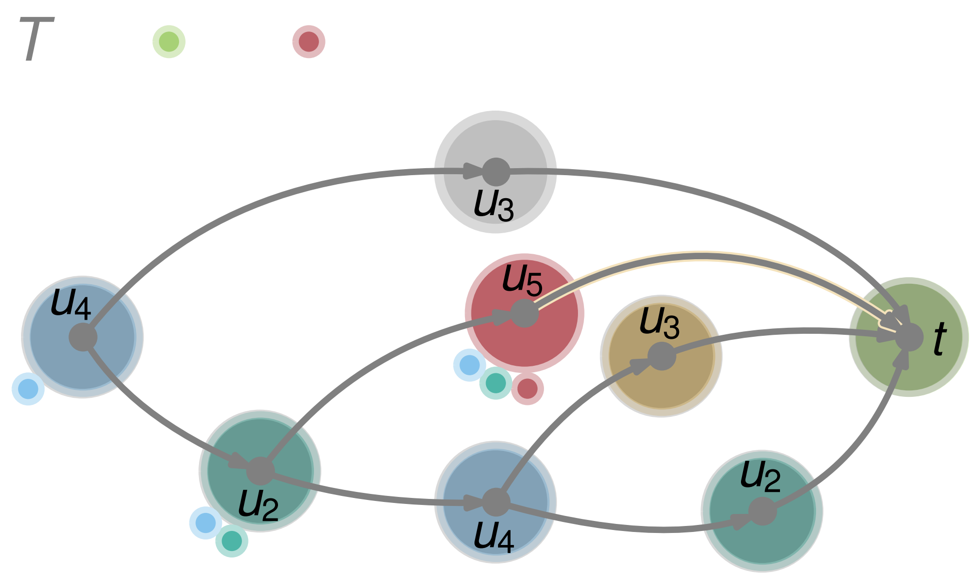


But: n colors $\Rightarrow$ not a polynomial time algorithm(?)

Rainbow Paths – FPT Algorithm

[Theorem 11; Uchizawa et al., 2013]

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

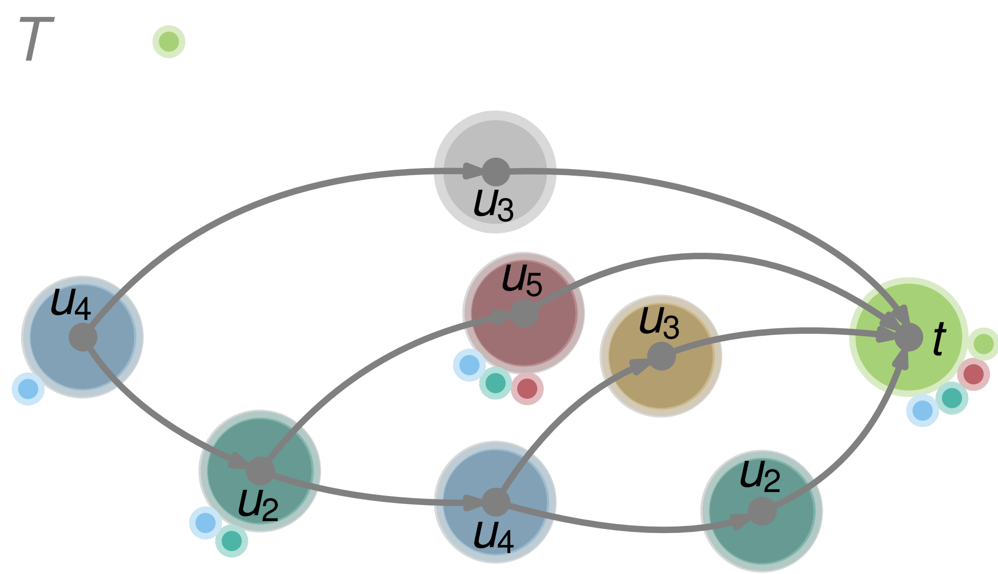


But: n colors $\Rightarrow$ not a polynomial time algorithm(?)

Rainbow Paths – FPT Algorithm

[Theorem 11; Uchizawa et al., 2013]

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



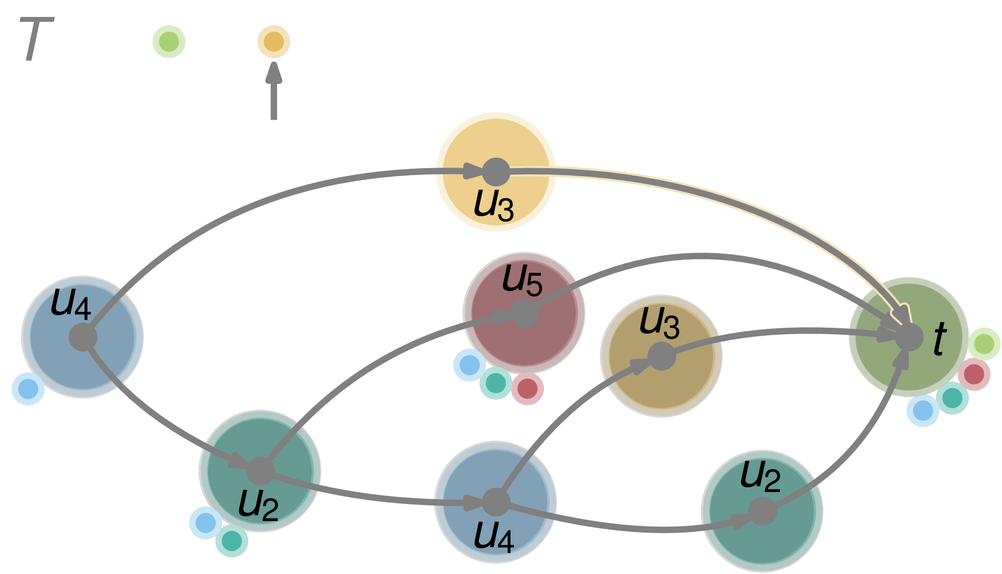
But: n colors $\Rightarrow$ not a polynomial time algorithm(?)

Institute of Theoretical Informatics
Algorithmics Group

Rainbow Paths – FPT Algorithm

[Theorem 11; Uchizawa et al., 2013]

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



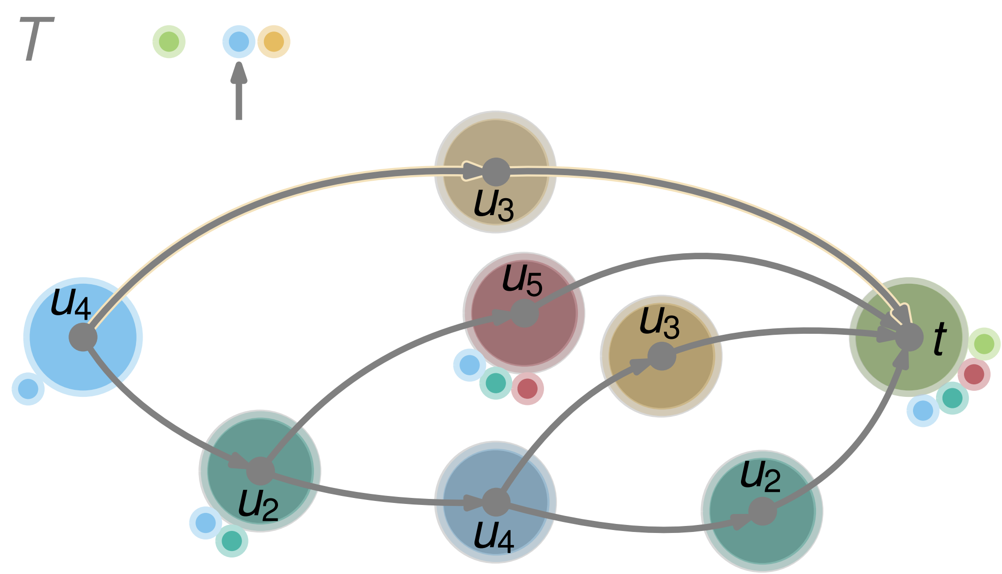
But: n colors $\Rightarrow$ not a polynomial time algorithm(?)

Institute of Theoretical Informatics
Algorithmics Group

Rainbow Paths – FPT Algorithm

[Theorem 11; Uchizawa et al., 2013]

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



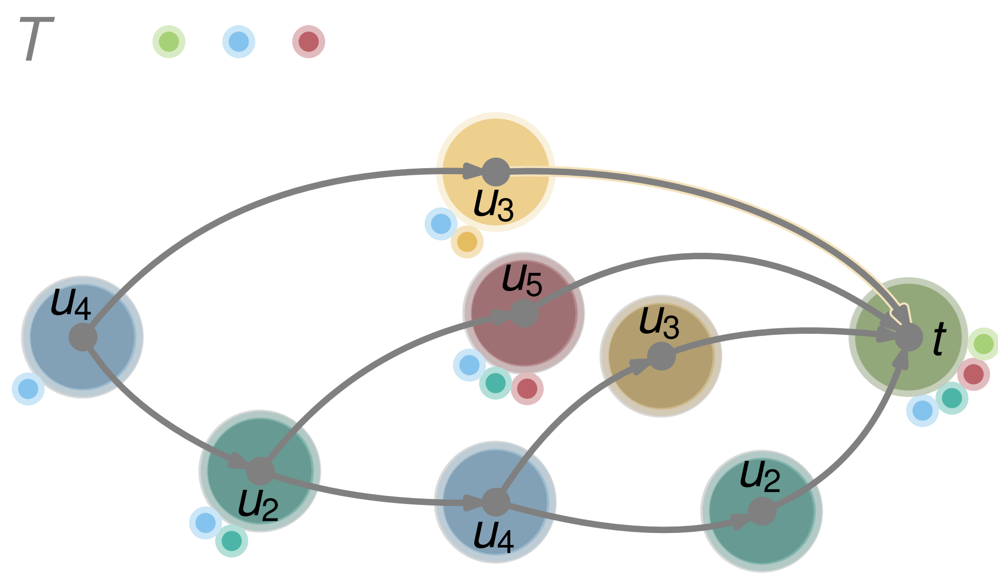
But: n colors $\Rightarrow$ not a polynomial time algorithm(?)

Institute of Theoretical Informatics
Algorithmics Group

Rainbow Paths – FPT Algorithm

[Theorem 11; Uchizawa et al., 2013]

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

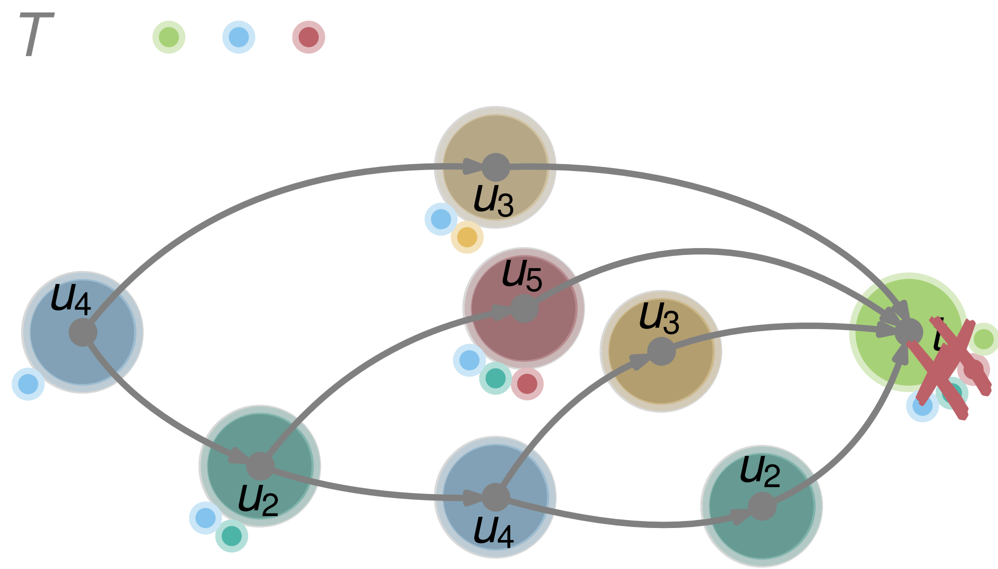


But: n colors $\Rightarrow$ not a polynomial time algorithm(?)

Rainbow Paths – FPT Algorithm

[Theorem 11; Uchizawa et al., 2013]

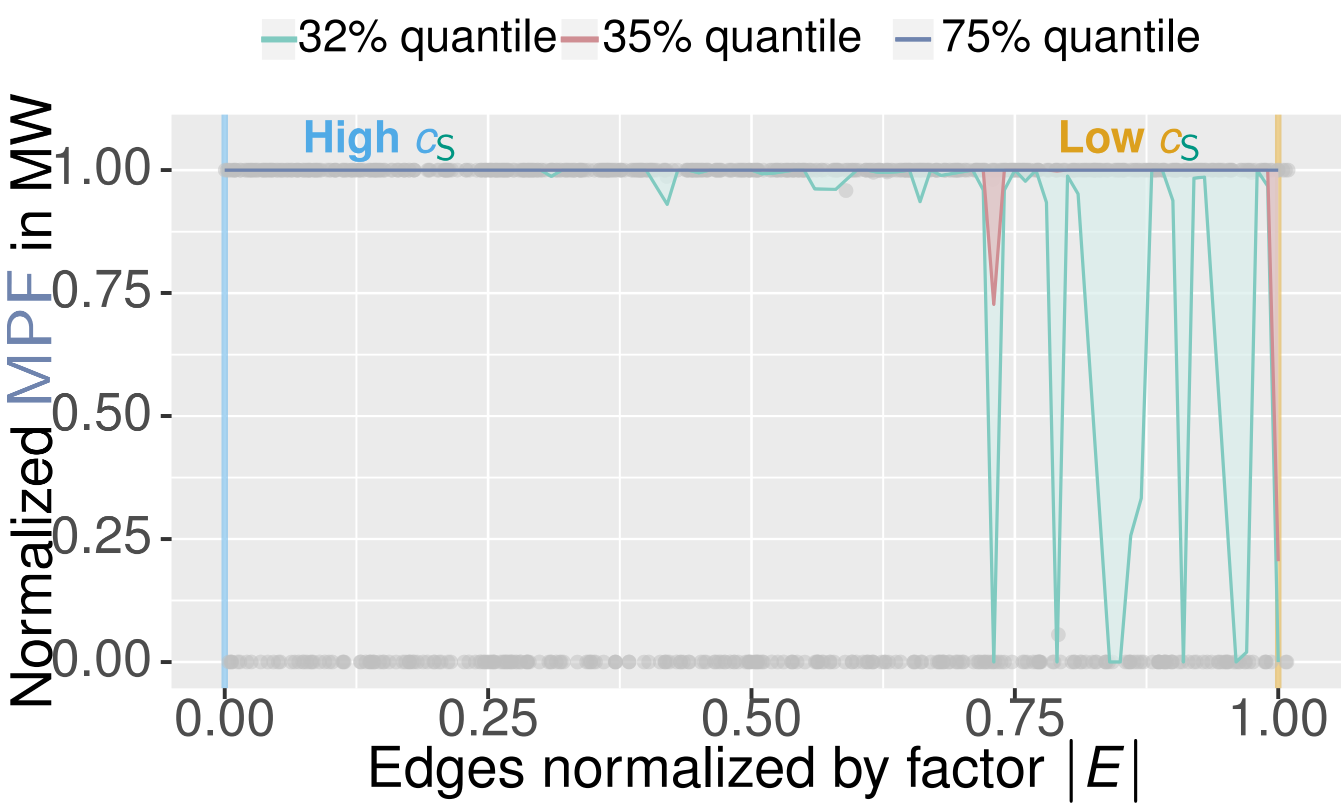
- FPT algorithm w.r.t. the number of colors



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* $c_S : E \to \mathbb{R}_{\geq 0}$ is defined by

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

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

0.00 0.25 0.50 0.75 1.00

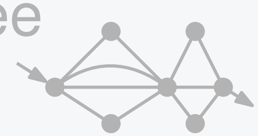
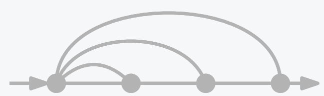
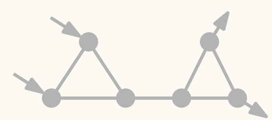
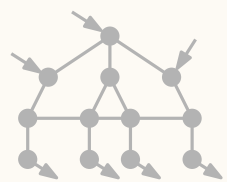
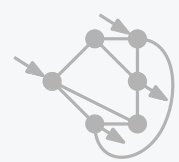
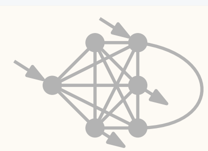
Edges normalized by factor $|E|$

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

Institute of Theoretical Informatics
Algorithmics Group

Overview of the MTSF Results

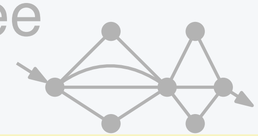
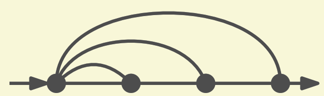
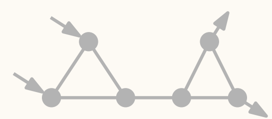
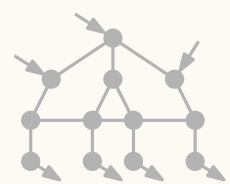
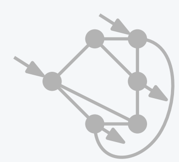
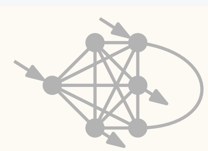
complexity

	Graph Structure		Complexity	Algorithm
one generator, one load	penrose-minor-free graphs		polynomial-time solvable	DTP ✓
	series-parallel graphs		NP-hard	✗
arbitrary generators, arbitrary loads	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]	✗
$\lvert V_G \rvert = 2$, $\lvert V_C \rvert = 2$	arbitrary graphs		non-APX [Lehmann et al., 2014]	✗

Institute of Theoretical Informatics
Algorithmics Group

Overview of the MTSF Results

complexity

	Graph Structure		Complexity	Algorithm
one generator, one load	penrose-minor-free graphs		polynomial-time solvable	DTP ✓
	series-parallel graphs		NP-hard	✗
arbitrary generators, arbitrary loads	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]	✗
$\lvert V_G \rvert = 2$, $\lvert V_C \rvert = 2$	arbitrary graphs		non-APX [Lehmann et al., 2014]	✗

Institute of Theoretical Informatics
Algorithmics Group

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 $\sum_{j=1}^{n} w_j x_j = k$?

Institute of Theoretical Informatics
Algorithmics Group

Subset Sum Problem (SSP)

Decision Problem Subset Sum (SSP)

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

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

Example Instance

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

Institute of Theoretical Informatics
Algorithmics Group

Subset Sum Problem (SSP)

Karlsruhe Institute of Technology

Decision Problem Subset Sum (SSP)

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

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

Example Instance

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

Solution

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

Institute of Theoretical Informatics
Algorithmics Group

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

[Kocuk et al., 2016]

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

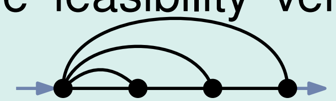
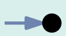
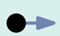
The feasibility version of OTS is NP-complete even if $\mathcal{N} = (G, V_G, V_C, \mathrm{cap}, b, \underline{d})$ is a series-parallel graph with one generator $|V_G| = 1$, and one consumer $|V_C| = 1$.

Institute of Theoretical Informatics
Algorithmics Group

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 $\mathcal{N} = (G, V_G, V_C, \mathrm{cap}, b, \underline{d})$ is a  with one generator  , and one consumer  .

Institute of Theoretical Informatics
Algorithmics Group

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

[Kocuk et al., 2016]

Karlsruhe Institute of Technology

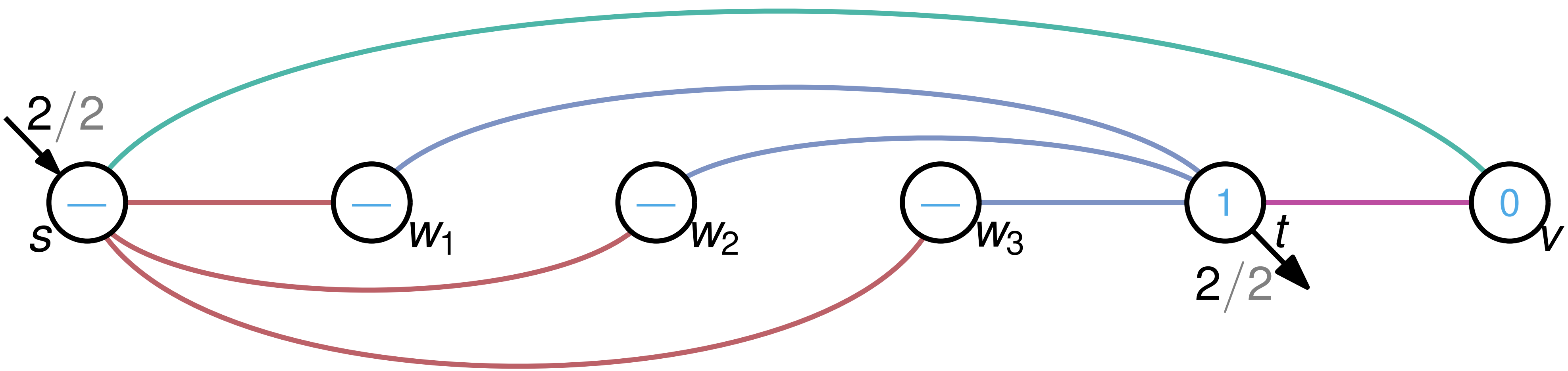
Instance:

vertices	$n+3$ vertices $\{0, 1, \ldots, n, n+1, n+2\}$
edges	$(0, i)$, $(i, n+1)$, $(n+1, n+2)$, $(0, n+2)$. $\quad \forall i \in \{1, \ldots, n\}$
capacities	$\frac{w_i}{k}$ for $(0, i), (i, n+1)$, 1 for $(n+1, n+2), (0, n+2)$, $\quad \forall i \in \{1, \ldots, n\}$
susceptances	$2w_i$ for $(0, i), (i, n+1)$, 1 for $(n+1, n+2)$, $\frac{k}{k+1}$ for $(0, n+2)$ $\quad \forall i \in \{1, \ldots, n\}$
generation / load	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]

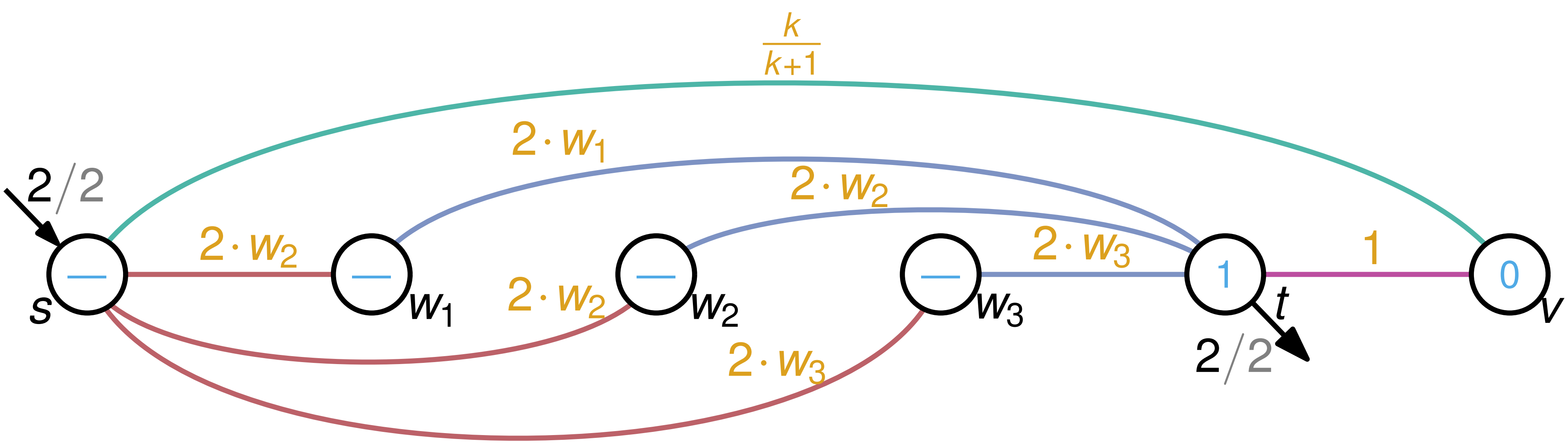
- **Instance:** $n = 3$, $W = \{1, 2, 3\}$, and $k = 3$.



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

[Kocuk et al., 2016]

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



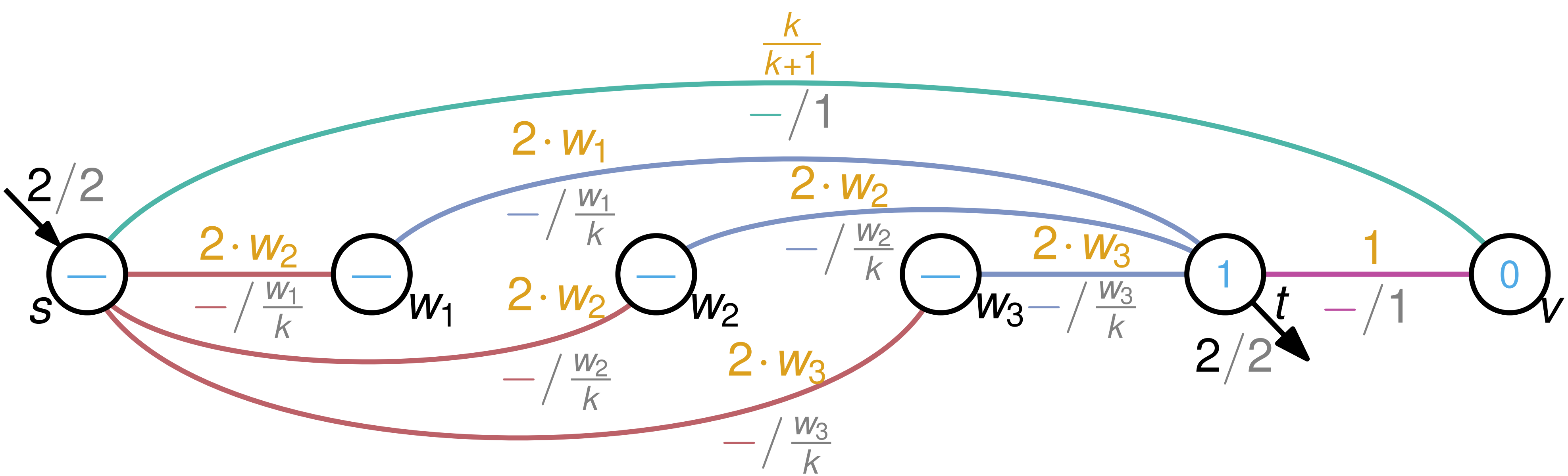
Institute of Theoretical Informatics
Algorithmics Group

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

[Kocuk et al., 2016]

Karlsruhe Institute of Technology

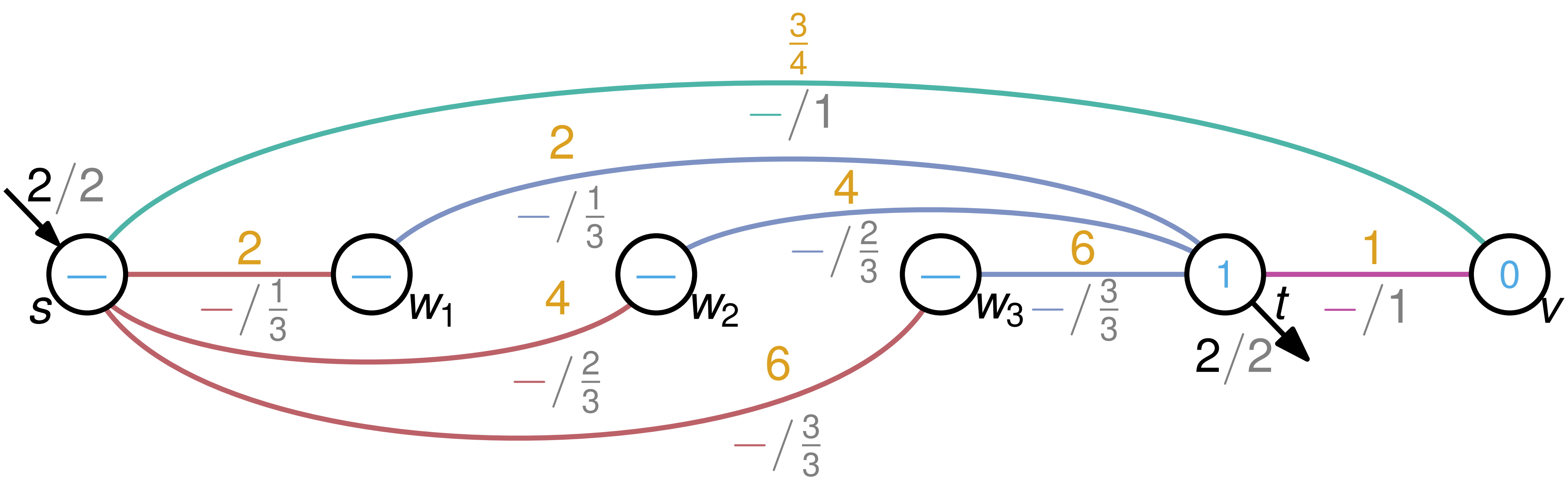
- **Instance:** $n = 3$, $W = \{1, 2, 3\}$, and $k = 3$.



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

[Kocuk et al., 2016]

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

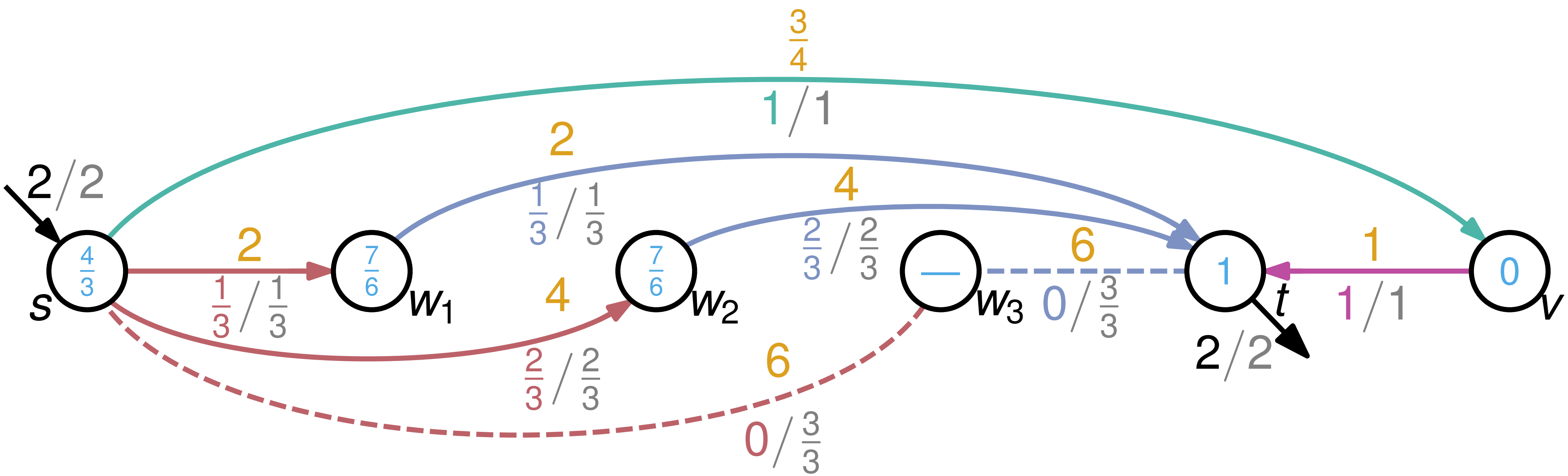


OTS is NP-complete on Series-parallel Graphs Single-Source-Sink

[Kocuk et al., 2016]

Karlsruhe Institute of Technology

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



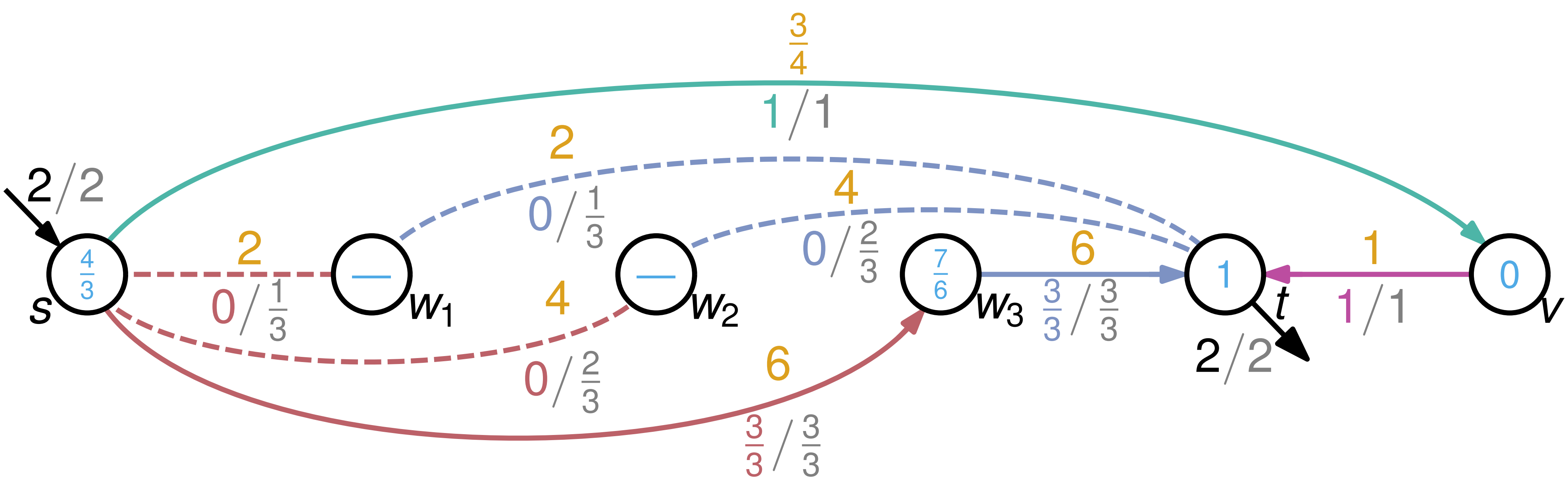
Institute of Theoretical Informatics
Algorithmics Group

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

[Kocuk et al., 2016]

Karlsruhe Institute of Technology

- **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]

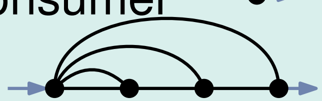
The k-MTSF problem is NP-complete even if there is only one generator $|V_G| = 1$ and one consumer $|V_C| = 1$ in the network $\mathcal{N} = (G, V_G, V_C, \text{cap}, b, \underline{d})$ where the underlying graph G is series-parallel and all edge capacities are $\text{cap}(e) = 1$ for all e in E.

Institute of Theoretical Informatics
Algorithmics Group

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

Karlsruhe Institute of Technology

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

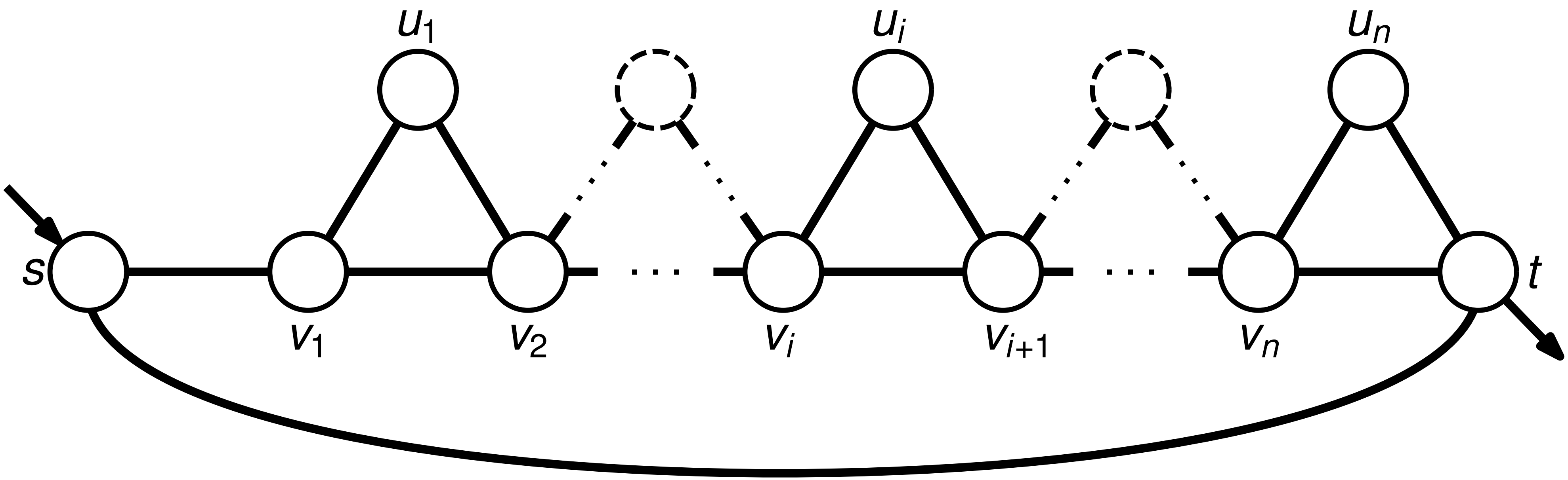
The k-MTSF problem is NP-complete even if there is only one generator and one consumer in the network $\mathcal{N} = (G, V_G, V_C, \text{cap}, b, \underline{d})$ where the underlying graph G is  and all edge capacities are $\text{cap}(e) = 1$ for all e in E.

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

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

$|V_G| := |V_C| := 1$

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

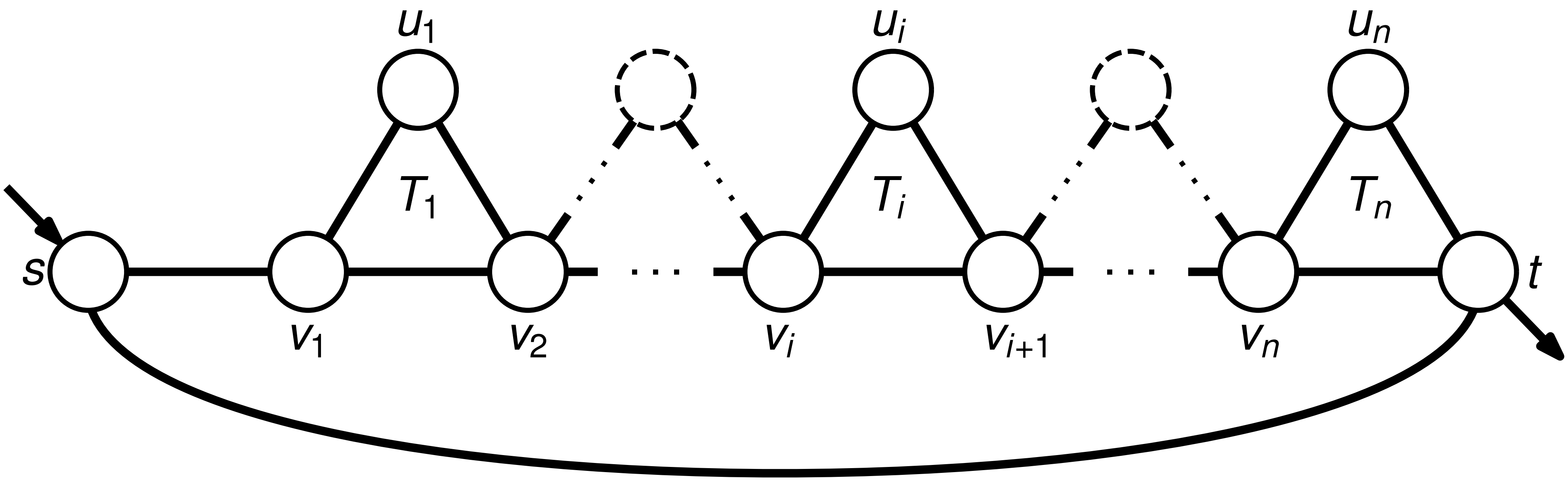


k-MTSF is NP-complete on Series-parallel Graphs
Single-Source-Sink and Unit-Capacities [Grastien et al., 2018]

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

$|V_G| := |V_C| := 1$

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

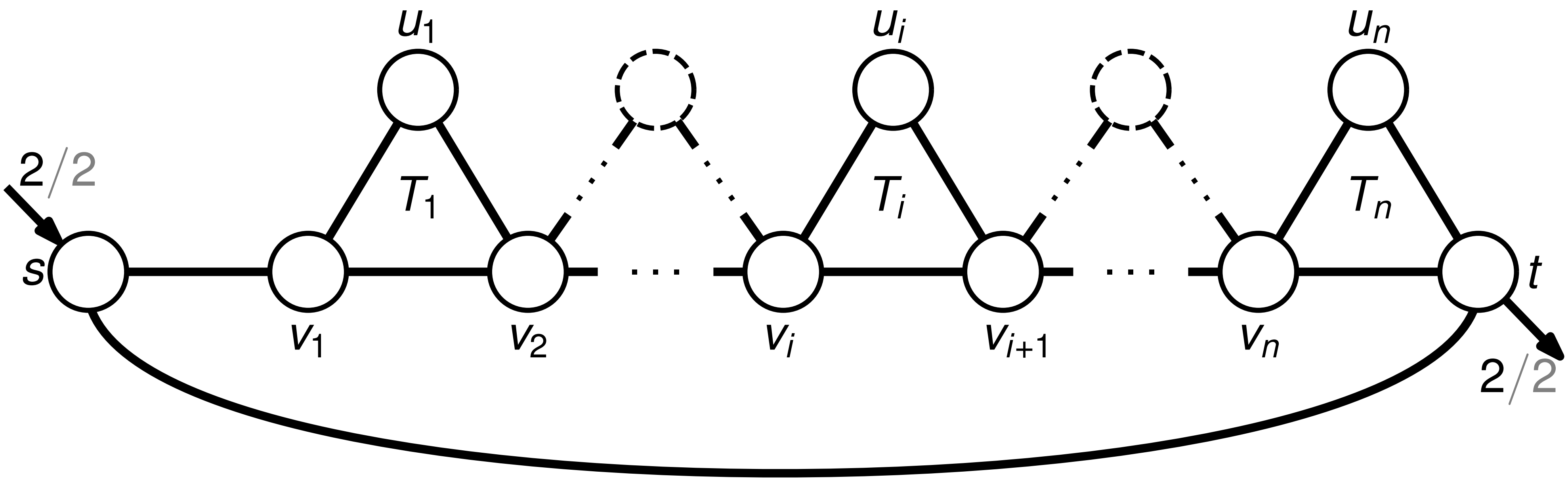


k-MTSF is NP-complete on Series-parallel Graphs
Single-Source-Sink and Unit-Capacities [Grastien et al., 2018]

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

$|V_G| := |V_C| := 1$

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



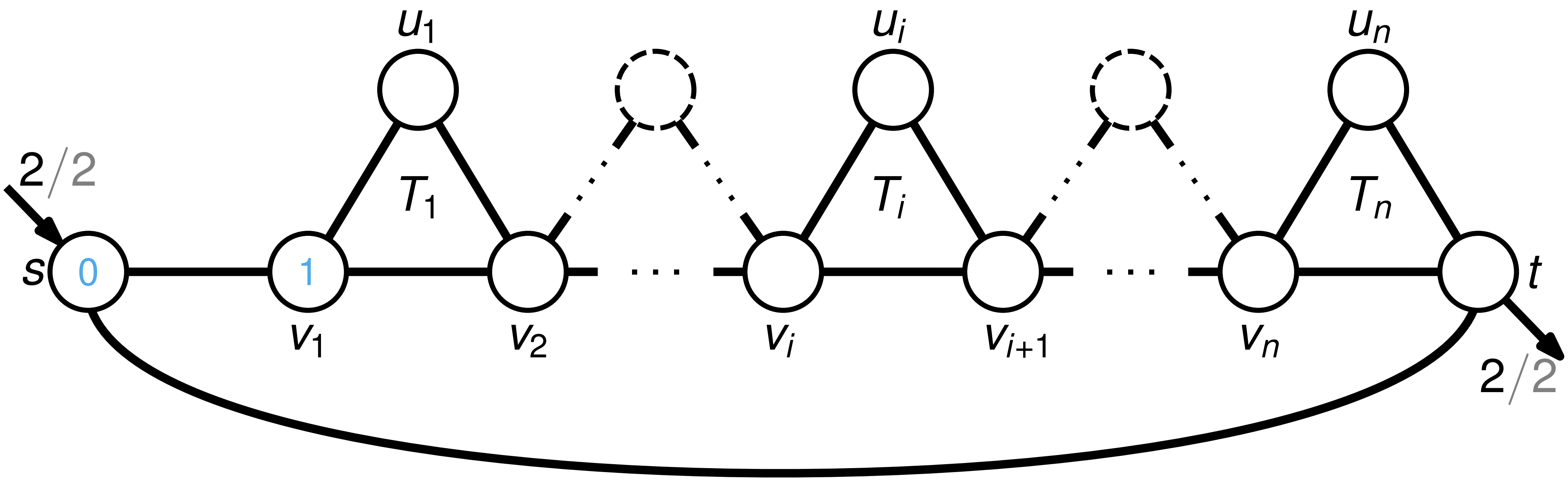
Institute of Theoretical Informatics
Algorithmics Group

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

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

$|V_G| := |V_C| := 1$

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



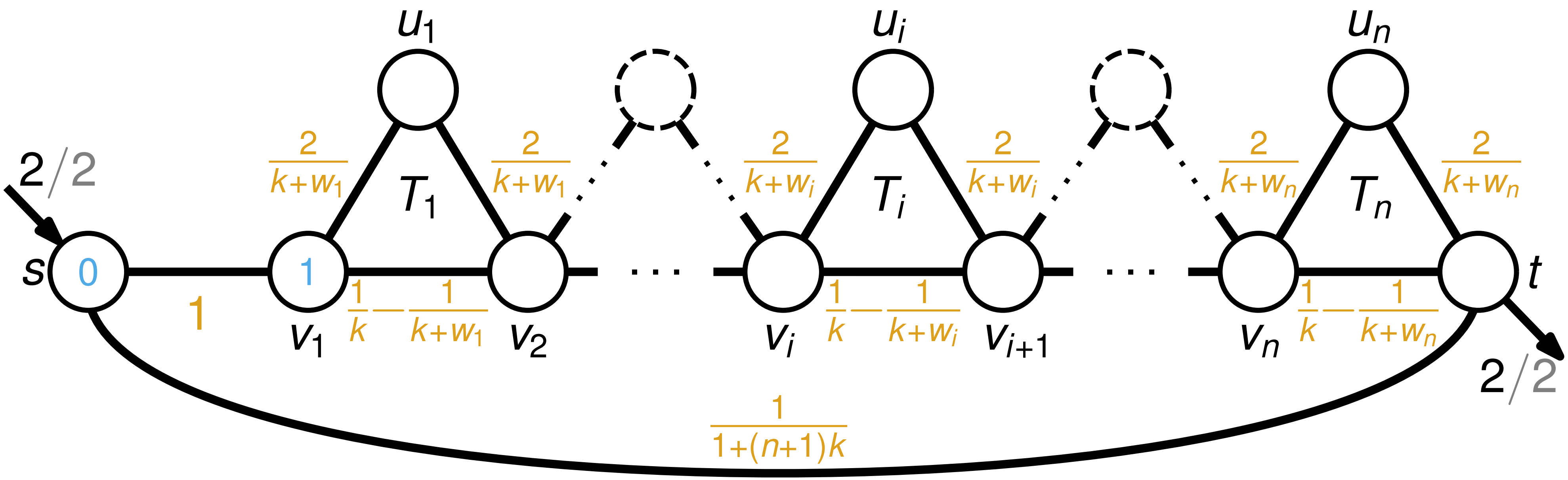
Institute of Theoretical Informatics
Algorithmics Group

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

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

$|V_G| := |V_C| := 1$

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



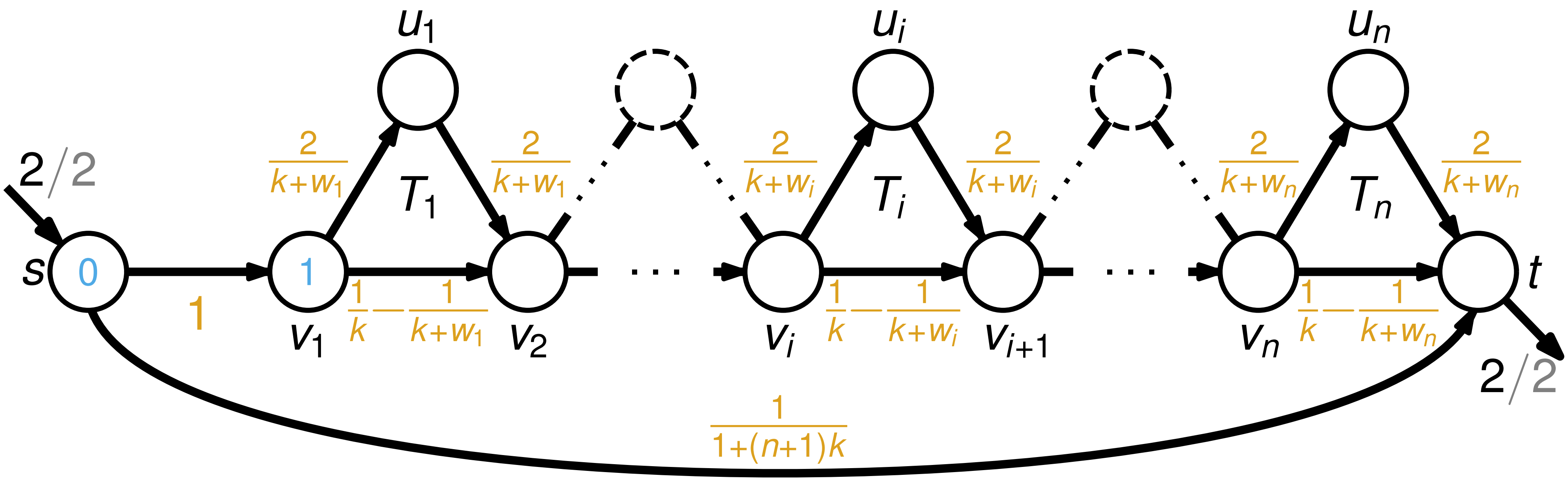
k-MTSF is NP-complete on Series-parallel Graphs
Single-Source-Sink and Unit-Capacities [Grastien et al., 2018]

Karlsruhe Institute of Technology

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

$|V_G| := |V_C| := 1$

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

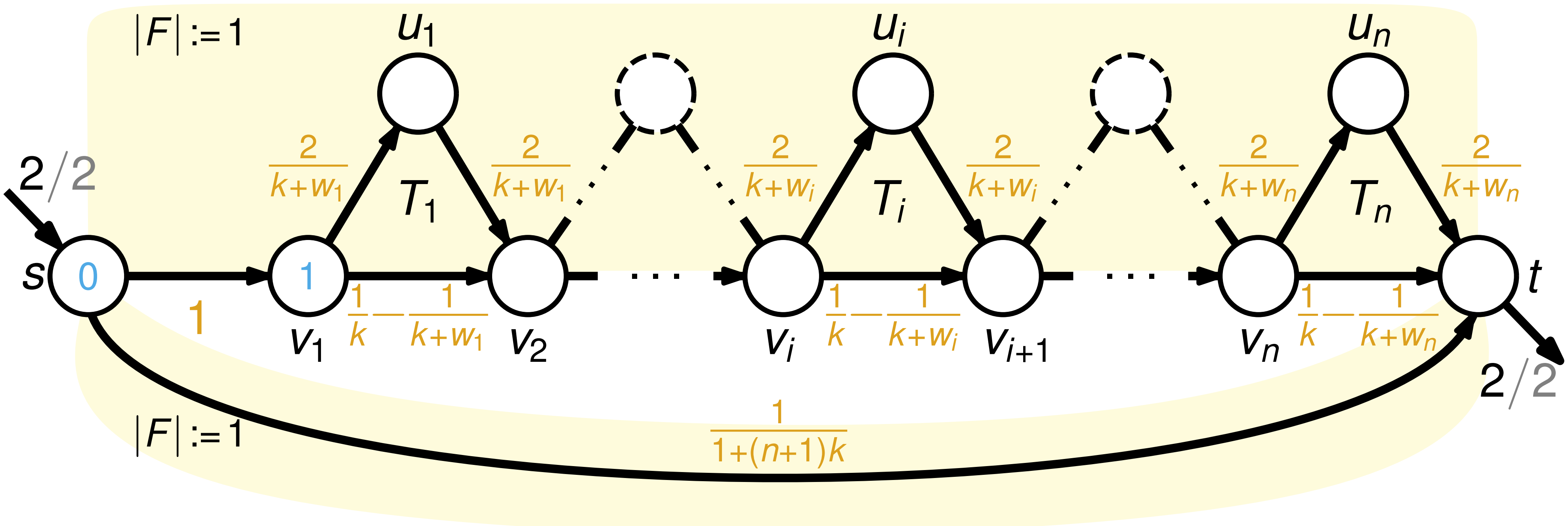


k-MTSF is NP-complete on Series-parallel Graphs
Single-Source-Sink and Unit-Capacities [Grastien et al., 2018]

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

$|V_G| := |V_C| := 1$

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

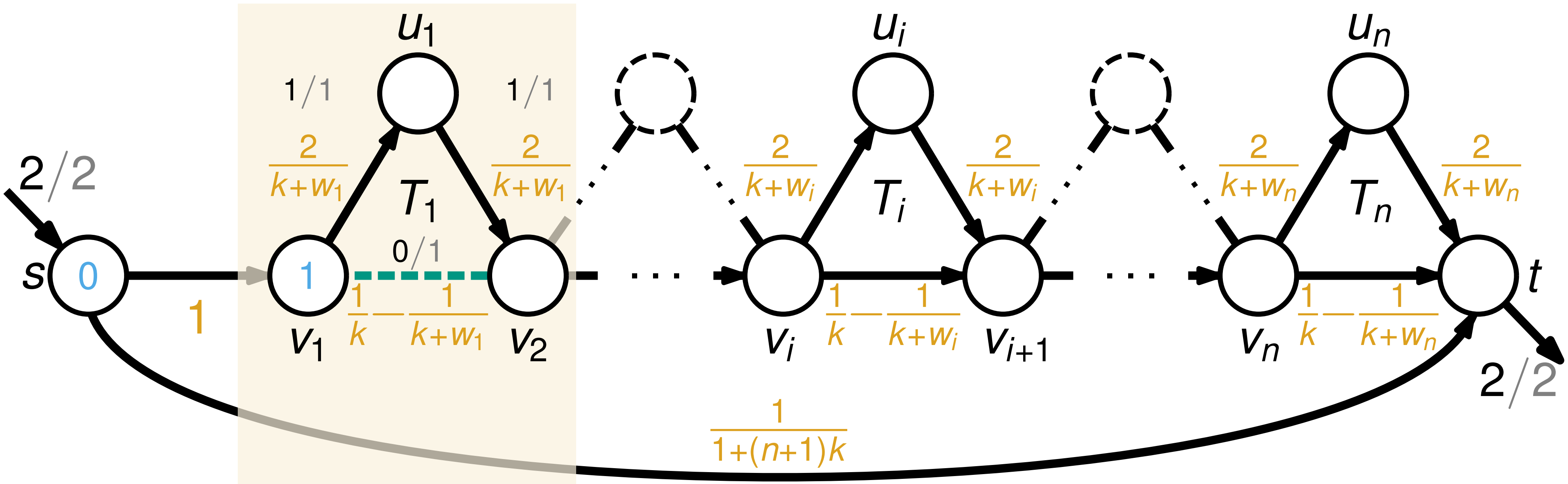


k-MTSF is NP-complete on Series-parallel Graphs
Single-Source-Sink and Unit-Capacities [Grastien et al., 2018]

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

$|V_G| := |V_C| := 1$

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



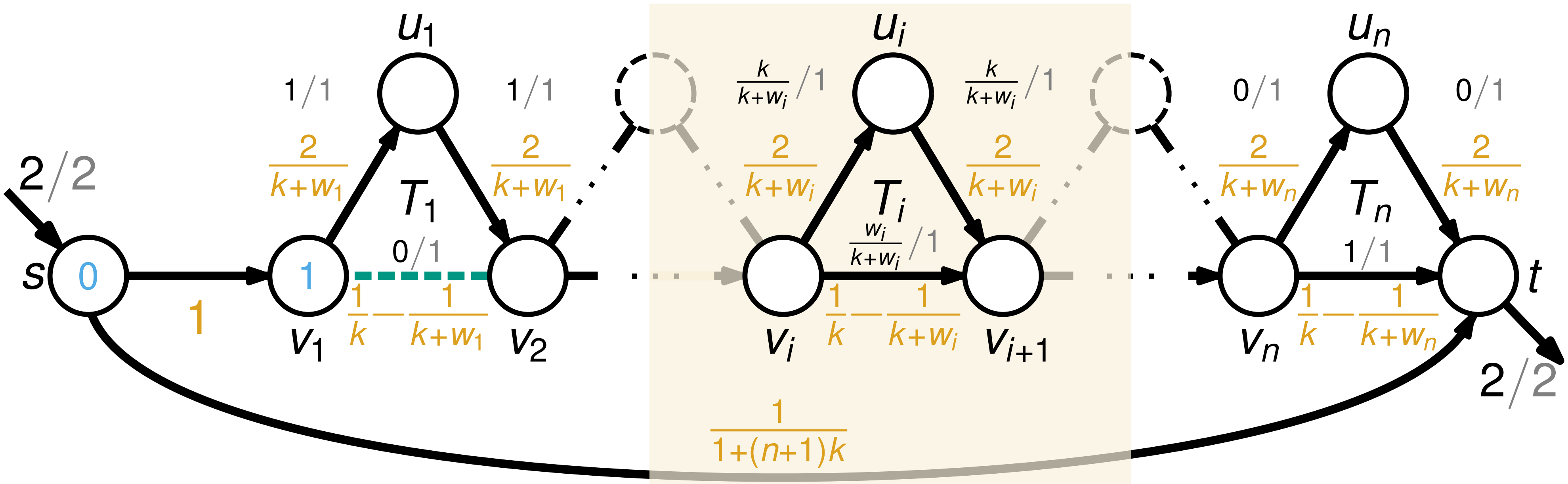
$$b_{\text{total}}(T_1) = \frac{1}{\frac{1}{\frac{2}{k+w_1}} + \frac{1}{\frac{2}{k+w_1}}} = \frac{1}{\frac{k+w_1}{2} + \frac{k+w_1}{2}} = \frac{1}{k+w_1}$$

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

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

$|V_G| := |V_C| := 1$

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



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

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

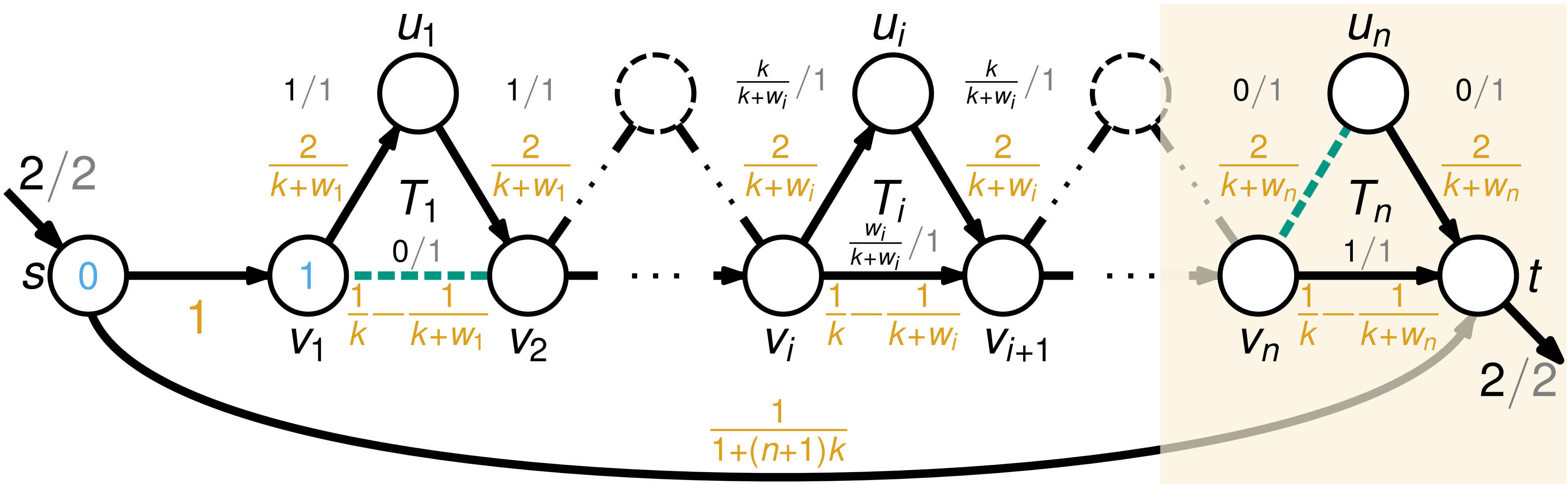
Institute of Theoretical Informatics
Algorithmics Group

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

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

$|V_G| := |V_C| := 1$

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



$$b_{\mathrm{total}}(T_1) = \frac{1}{\frac{1}{\frac{2}{k+w_1}} + \frac{1}{\frac{2}{k+w_1}}} = \frac{1}{\frac{k+w_1}{2} + \frac{k+w_1}{2}} = \frac{1}{k+w_1}$$

$$b_{\mathrm{total}}(T_i) = \frac{1}{\frac{1}{\frac{2}{k+w_i}} + \frac{1}{\frac{2}{k+w_i}}} + \frac{1}{k} - \frac{1}{k+w_i} = \frac{1}{k+w_i} + \frac{1}{k} - \frac{1}{k+w_i} = \frac{1}{k}$$

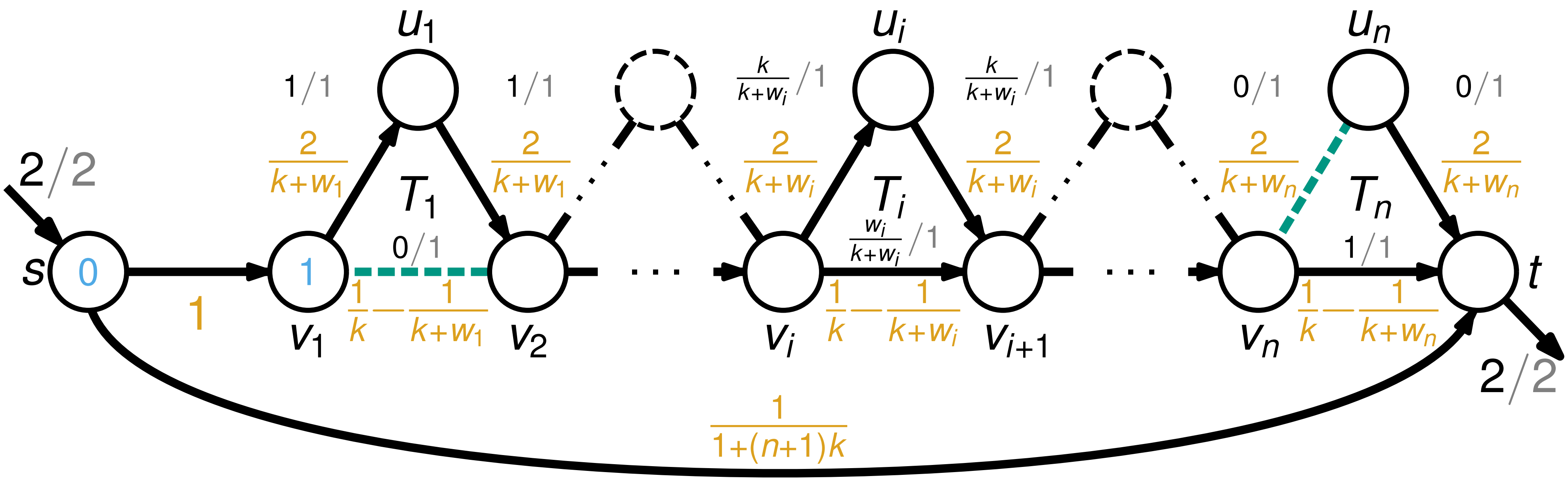
$$b_{\mathrm{total}}(T_n) = \frac{1}{k} - \frac{1}{k+w_n} = \frac{k+w_n}{k(k+w_n)} - \frac{k}{k(k+w_n)} = \frac{w_n}{k(k+w_n)}$$

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

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

$|V_G| := |V_C| := 1$

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



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

$\Delta\theta(v_1, v_2) = k + w_1$ $\quad$ $\Delta\theta(v_i, v_{i+1}) = k$ $\quad$ $\Delta\theta(v_n, t) = \frac{k(k+w_n)}{w_n} = \frac{k^2}{w_n} + k > 2k$

Institute of Theoretical Informatics
Algorithmics Group

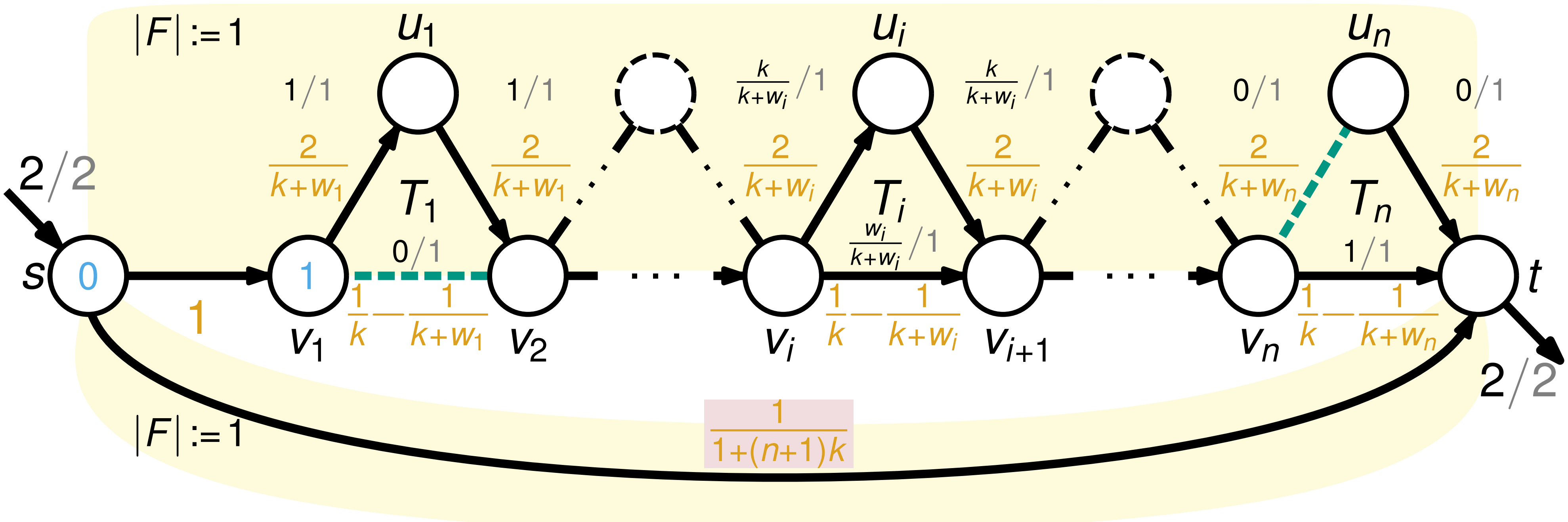
k-MTSF is NP-complete on Series-parallel Graphs
Single-Source-Sink and Unit-Capacities [Grastien et al., 2018]

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

$|V_G| := |V_C| := 1$

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

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



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

$\Delta\theta(v_1, v_2) = k + w_1$ $\qquad$ $\Delta\theta(v_i, v_{i+1}) = k$ $\qquad$ $\Delta\theta(v_n, t) = \frac{k(k+w_n)}{w_n} = \frac{k^2}{w_n} + k > 2k$

Institute of Theoretical Informatics
Algorithmics Group

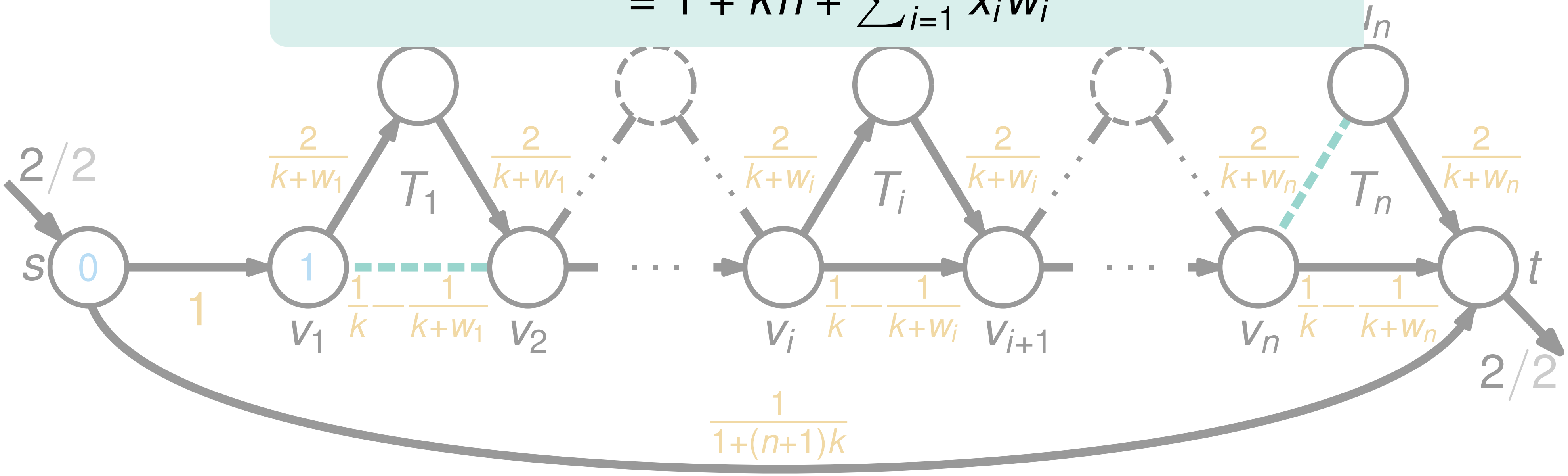
k-MTSF is NP-complete on Series-parallel Graphs
Single-Source-Sink and Unit-Capacities [Grastien et al., 2018]

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

$|V_G| := |V_C| := 1$

$W = \{w_1, w_2, \ldots$

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

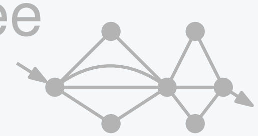
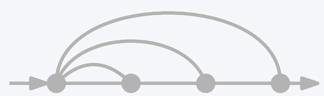
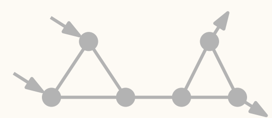
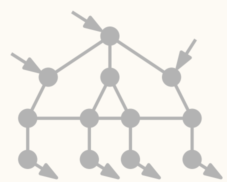
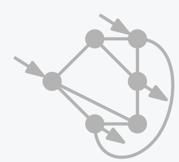
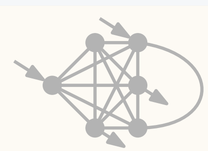


$b_{\mathrm{total}}(T_1) = \frac{1}{k+w_1}$ $\quad$ $b_{\mathrm{total}}(T_i) = \frac{1}{k}$ $\quad$ $b_{\mathrm{total}}(T_n) = \frac{w_n}{k(k+w_n)}$

$\Delta\theta(v_1, v_2) = k + w_1$ $\quad$ $\Delta\theta(v_i, v_{i+1}) = k$ $\quad$ $\Delta\theta(v_n, t) = \frac{k(k+w_n)}{w_n} = \frac{k^2}{w_n} + k > 2k$

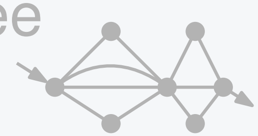
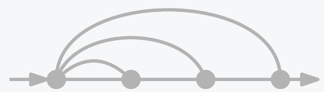
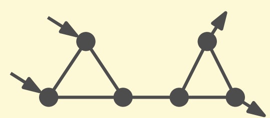
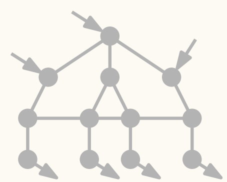
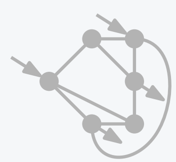
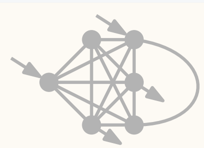
Overview of the MTSF Results

complexity

	Graph Structure		Complexity	Algorithm
one generator, one load	penrose-minor-free graphs		polynomial-time solvable	DTP ✓
	series-parallel graphs		NP-hard	✗
arbitrary generators, arbitrary loads	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]	✗
$\lvert V_G \rvert = 2$, $\lvert V_C \rvert = 2$	arbitrary graphs		non-APX [Lehmann et al., 2014]	✗

Overview of the MTSF Results

complexity

	Graph Structure	Complexity	Algorithm
one generator, one load	penrose-minor-free graphs 	polynomial-time solvable	DTP ✓
one generator, one load	series-parallel graphs 	NP-hard	✗
arbitrary generators, arbitrary loads	**cacti with max degree of 3** 	**NP-hard** [Lehmann et al., 2014]	**2-approx.** ✓
arbitrary generators, arbitrary loads	2-level trees 	NP-hard [Lehmann et al., 2014]	✗
	planar graphs with max degree of 3 	strongly NP-hard [Lehmann et al., 2014]	✗
$\lvert V_G \rvert = 2$, $\lvert V_C \rvert = 2$	arbitrary graphs 	non-APX [Lehmann et al., 2014]	✗

Institute of Theoretical Informatics
Algorithmics Group

Subset Sum Problem (SSP)

Karlsruhe Institute of Technology

Decision Problem Subset Sum (SSP)

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

Question: Is there a set of elements $x_1, x_2, \ldots, x_n \in \{0, 1\}$ 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 $W = \{w_1, w_2, \ldots, w_n\}$ with $w_i \in \mathbb{N}$ and $k \in \mathbb{N}$.

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

Example Instance

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

Institute of Theoretical Informatics
Algorithmics Group

Subset Sum Problem (SSP)

Decision Problem Subset Sum (SSP)

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

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

Example Instance

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

Solution

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

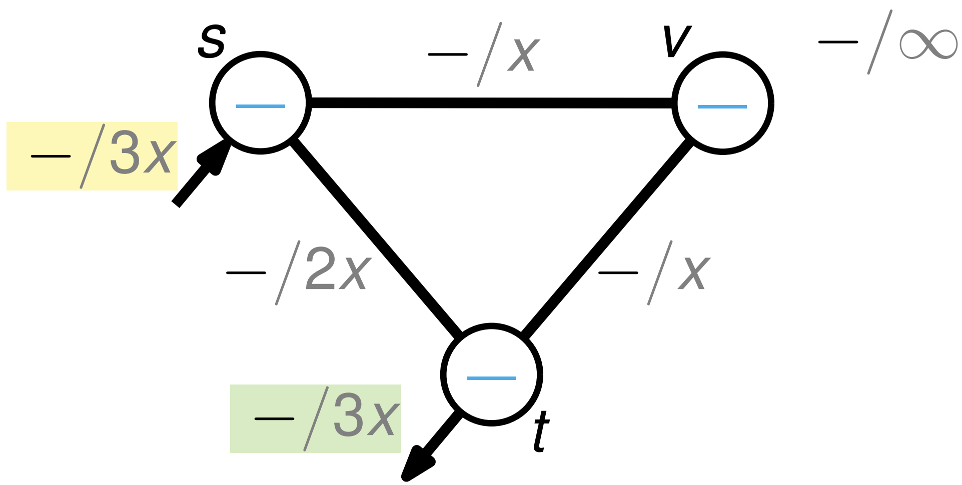
Institute of Theoretical Informatics
Algorithmics Group

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

Switching choice network SCN

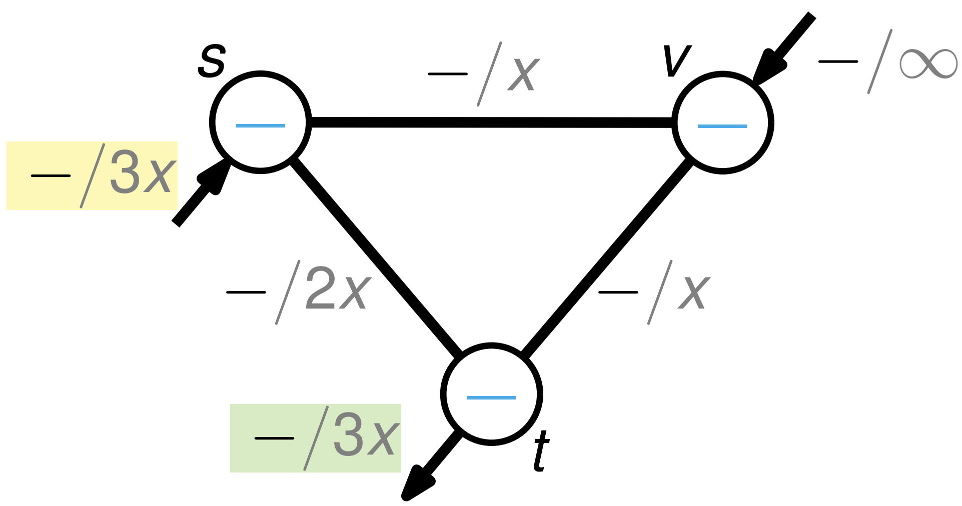


MTSF on Cacti is NP-hard

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

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

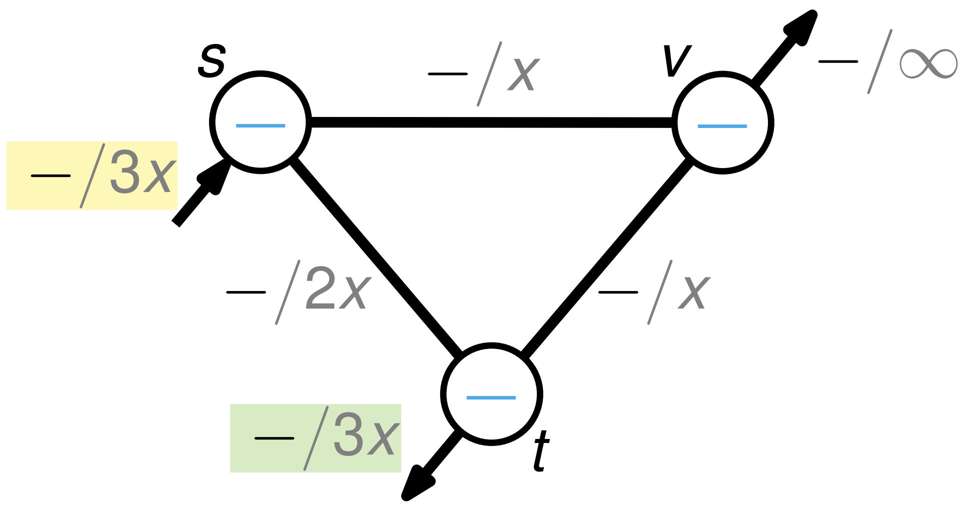


MTSF on Cacti is NP-hard

[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
  $\mathrm{SCN}_{\ell,v} = \left(\{s, t, v\}, E, \underline{d} := 3\ell, \overline{d} := 3\ell, \overline{x} := 3\ell, \mathrm{cap}\right)$

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



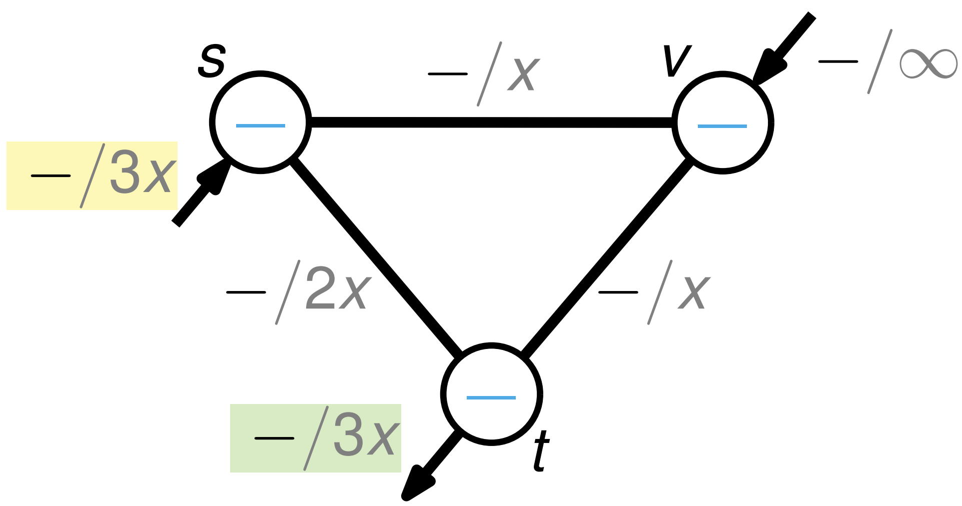
Institute of Theoretical Informatics
Algorithmics Group

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
  $\mathrm{SCN}_{\ell,v} = \left(\{s,t,v\}, E, \underline{d} := 3\ell, \overline{d} := 3\ell, \overline{x} := 3\ell, \mathrm{cap}\right)$

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



Institute of Theoretical Informatics
Algorithmics Group

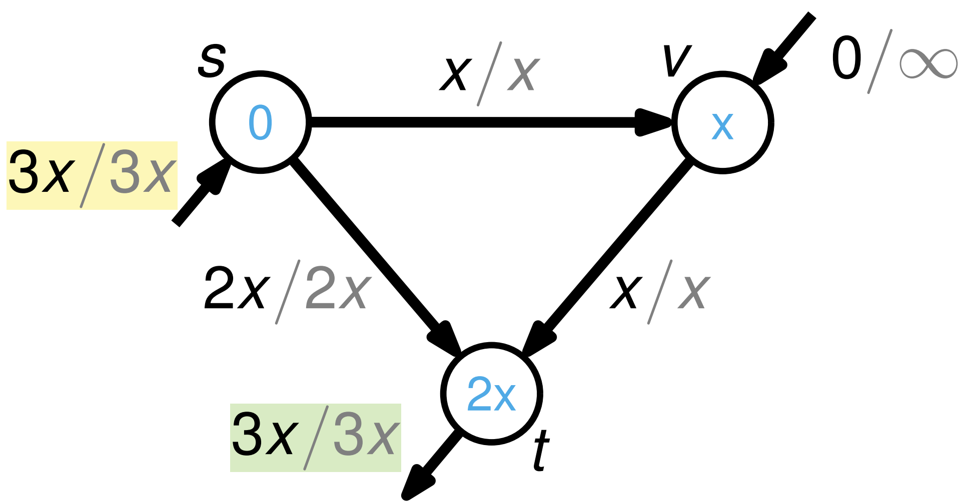
MTSF on Cacti is NP-hard

[Lehmann et al., 2014]

Karlsruhe Institute of Technology

- 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
  $\mathrm{SCN}_{\ell,v} = \left(\{s,t,v\}, E, \underline{d} := 3\ell, \overline{d} := 3\ell, \overline{x} := 3\ell, \mathrm{cap}\right)$

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



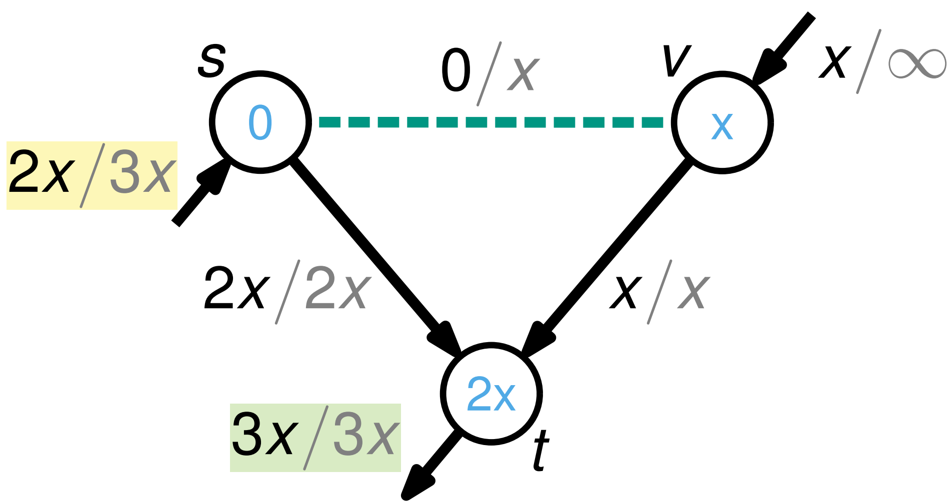
Institute of Theoretical Informatics
Algorithmics Group

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
  $\mathrm{SCN}_{\ell,v} = \left(\{s,t,v\}, E, \underline{d} := 3\ell, \overline{d} := 3\ell, \overline{x} := 3\ell, \mathrm{cap}\right)$

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



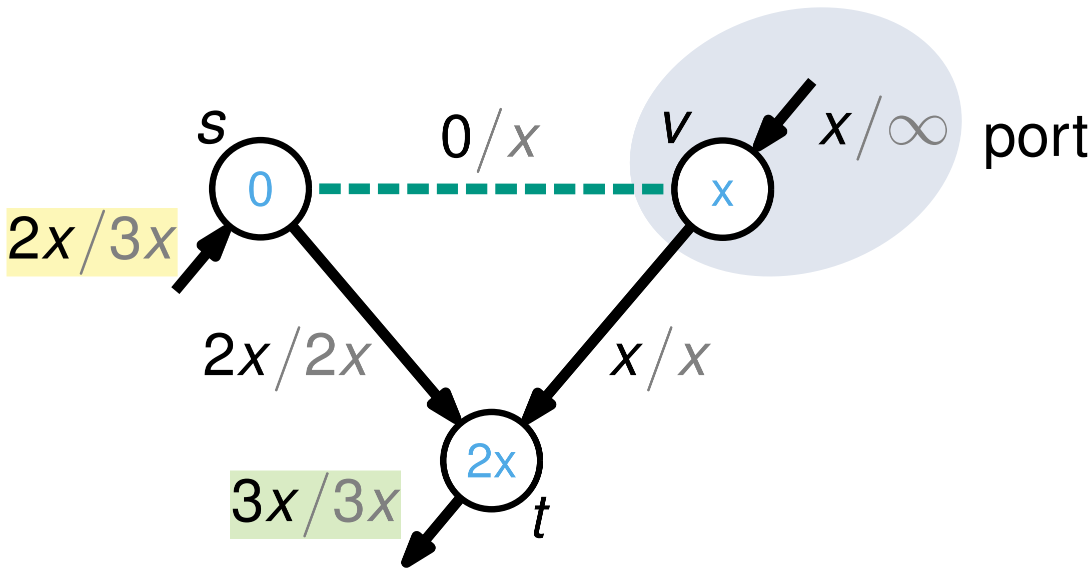
Institute of Theoretical Informatics
Algorithmics Group

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
  $\mathrm{SCN}_{\ell,v} = \left(\{s, t, v\}, E, \underline{d} := 3\ell, \overline{d} := 3\ell, \overline{x} := 3\ell, \mathrm{cap}\right)$

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

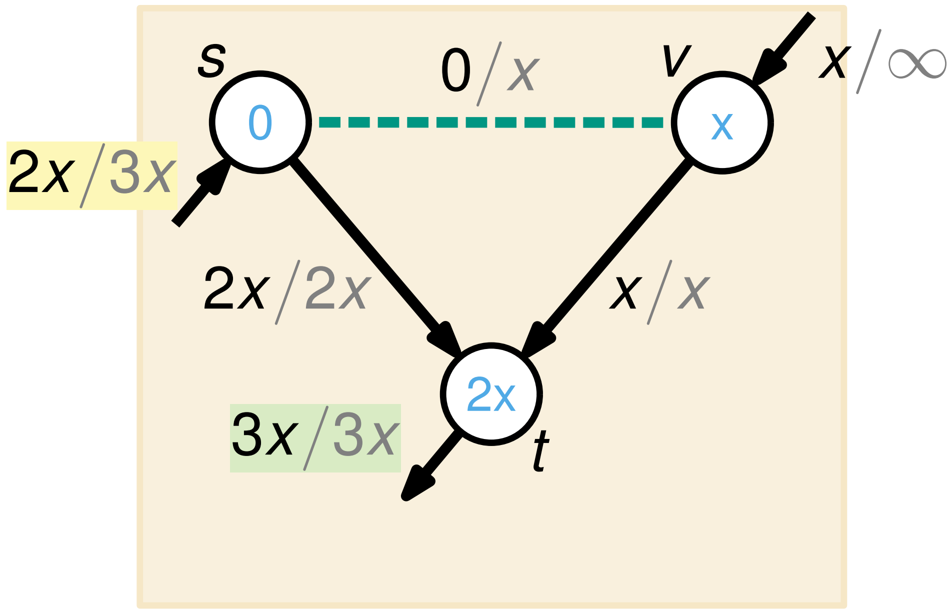


MTSF on Cacti is NP-hard

[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
  $\mathrm{SCN}_{\ell,v} = \left(\{s, t, v\}, E, \underline{d} := 3\ell, \overline{d} := 3\ell, \overline{x} := 3\ell, \mathrm{cap}\right)$

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



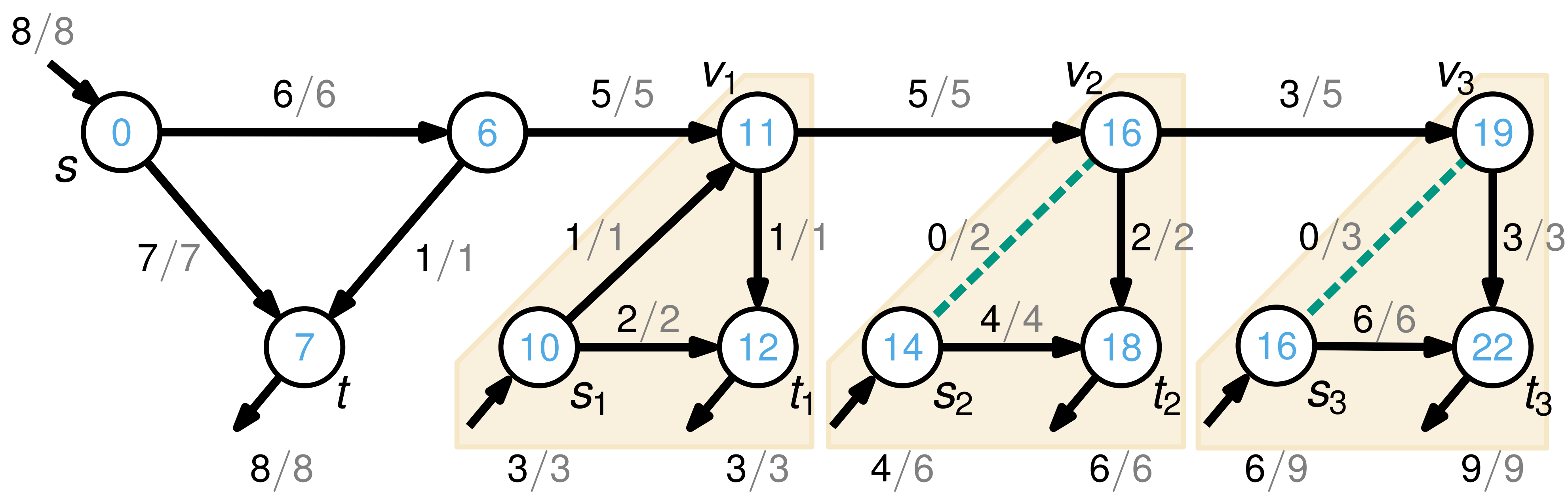
Institute of Theoretical Informatics
Algorithmics Group

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
  $\mathrm{SCN}_{\ell,v} = \left(\{s,t,v\}, E, \underline{d} := 3\ell, \overline{d} := 3\ell, \overline{x} := 3\ell, \mathrm{cap}\right)$

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



2-approximation on Cacti

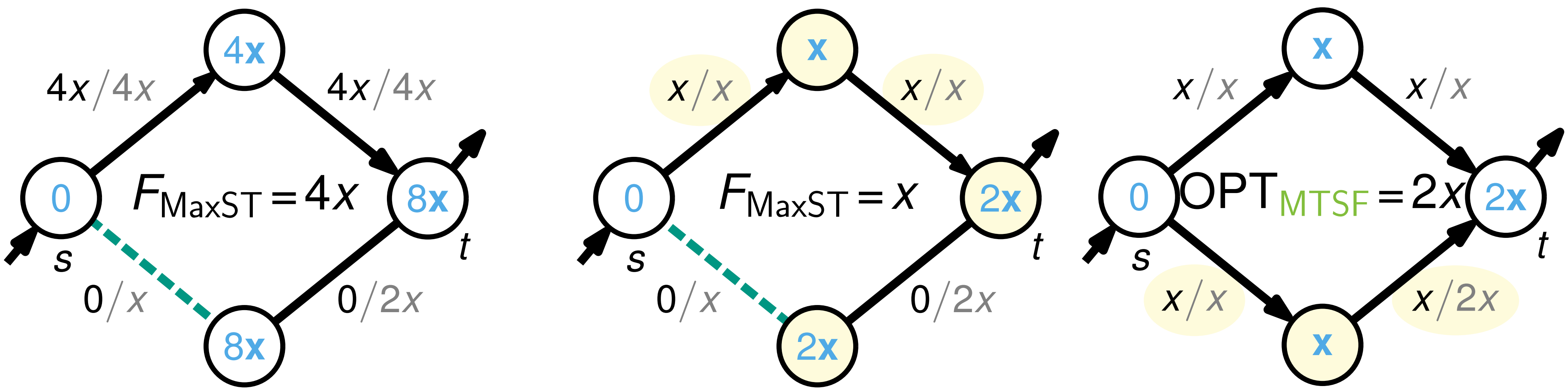
Description

- 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

⇔ 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 $\mathcal{N} = (G, V_G, V_C, \mathrm{cap}, b)$.
Result: $\mathrm{OPT}_{\mathrm{MPF}}(\mathcal{N} - S)$, and switched edges S.

$S = \emptyset$;
$C = \mathtt{dfs}(\mathcal{N})$;
for $c \in C$ **do**
    $S = S \cup \{\arg\min_{\forall e \in c}(\mathrm{cap}(e))\}$;
end
return $(\mathrm{OPT}_{\mathrm{MF}}(\mathcal{N} - S), S)$;

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

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

PROOF.

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

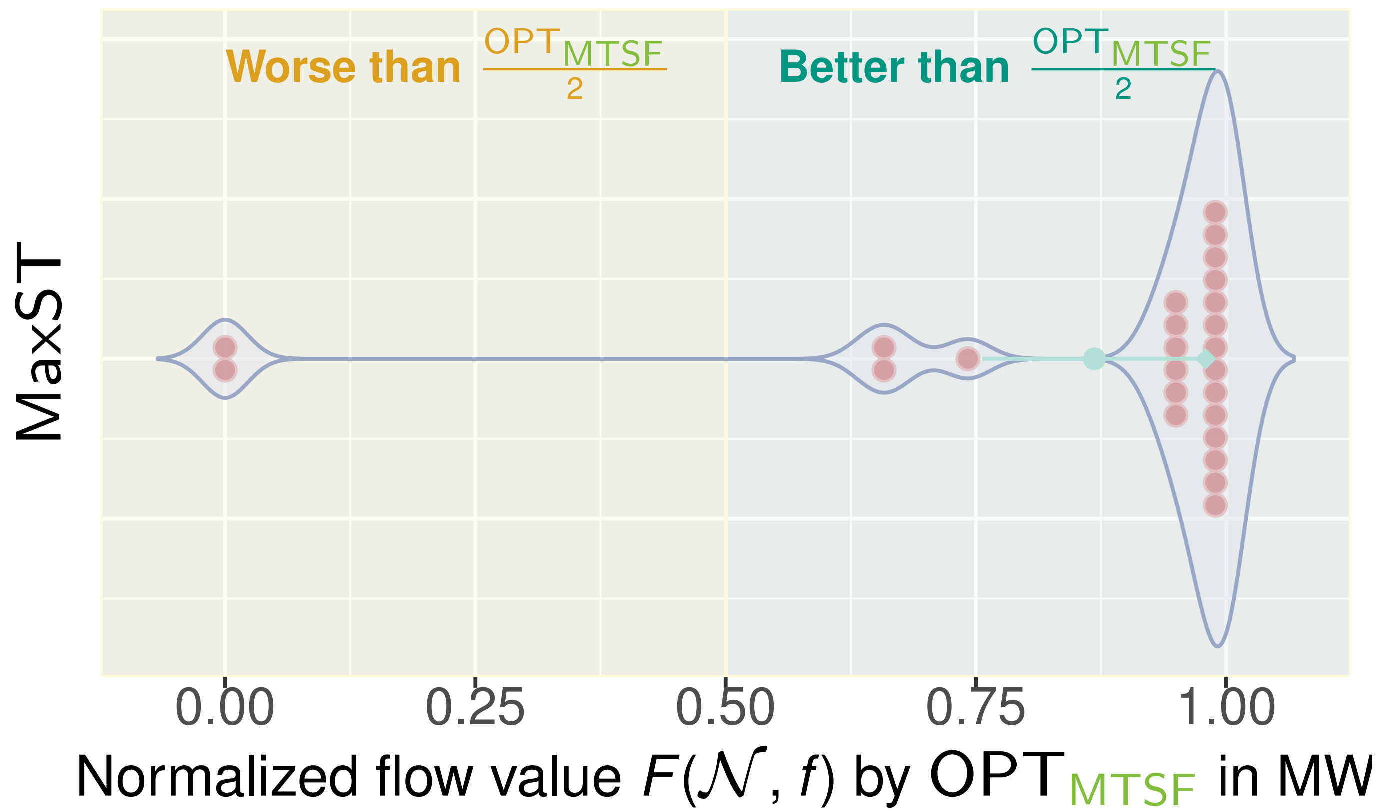
$$|f(e_{\min})| \leq \frac{1}{2}\,\mathrm{cap}(e_{\min}) \leq \frac{1}{2}\,\mathrm{cap}(e),$$

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

□

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

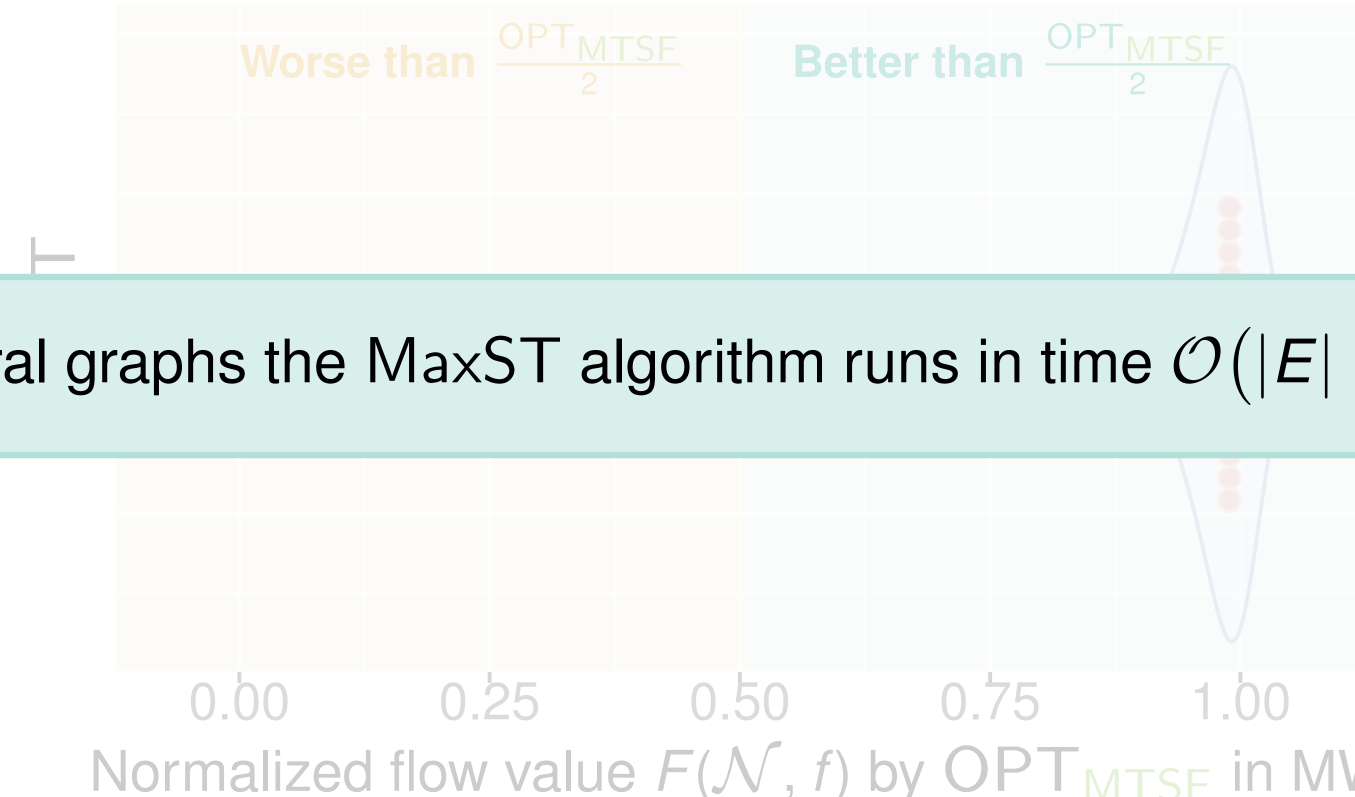


MaxST on **general graphs** is in most cases very close to the $\mathrm{OPT}_{\mathrm{MTSF}}$.

Institute of Theoretical Informatics
Algorithmics Group

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

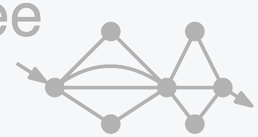
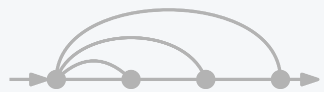
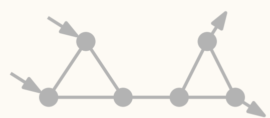
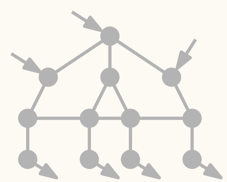
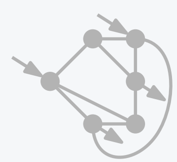
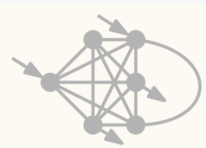


On general graphs the MaxST algorithm runs in time $\mathcal{O}\big(|E|\ \alpha(|E|, |V|)\big)$.

MaxST on **general graphs** is in most cases very close to the $\mathrm{OPT}_{\mathrm{MTSF}}$.

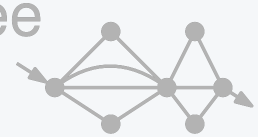
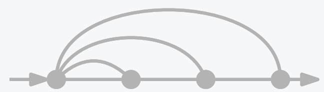
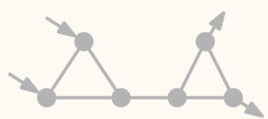
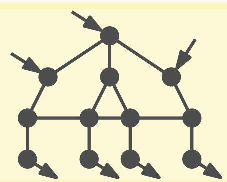
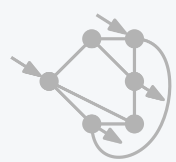
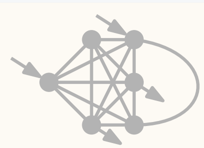
Overview of the MTSF Results

complexity

	Graph Structure		Complexity	Algorithm
one generator, one load	penrose-minor-free graphs		polynomial-time solvable	DTP ✓
	series-parallel graphs		NP-hard	✗
arbitrary generators, arbitrary loads	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]	✗
$\lvert V_G \rvert = 2$, $\lvert V_C \rvert = 2$	arbitrary graphs		non-APX [Lehmann et al., 2014]	✗

Institute of Theoretical Informatics
Algorithmics Group

Overview of the MTSF Results

	Graph Structure		Complexity	Algorithm
one generator, one load	penrose-minor-free graphs		polynomial-time solvable	DTP ✓
	series-parallel graphs		NP-hard	✗
arbitrary generators, arbitrary loads	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]	✗
$\lvert V_G \rvert = 2$, $\lvert V_C \rvert = 2$	arbitrary graphs		non-APX [Lehmann et al., 2014]	✗

complexity

Institute of Theoretical Informatics
Algorithmics Group

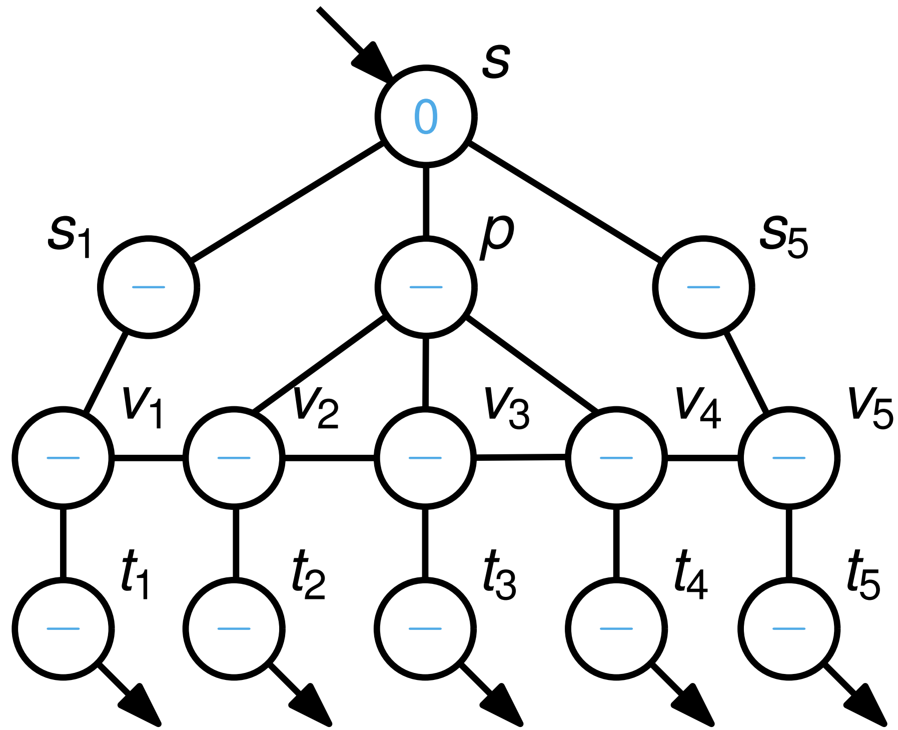
n-level Tree Graphs

[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 $t \in V_C$, 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]

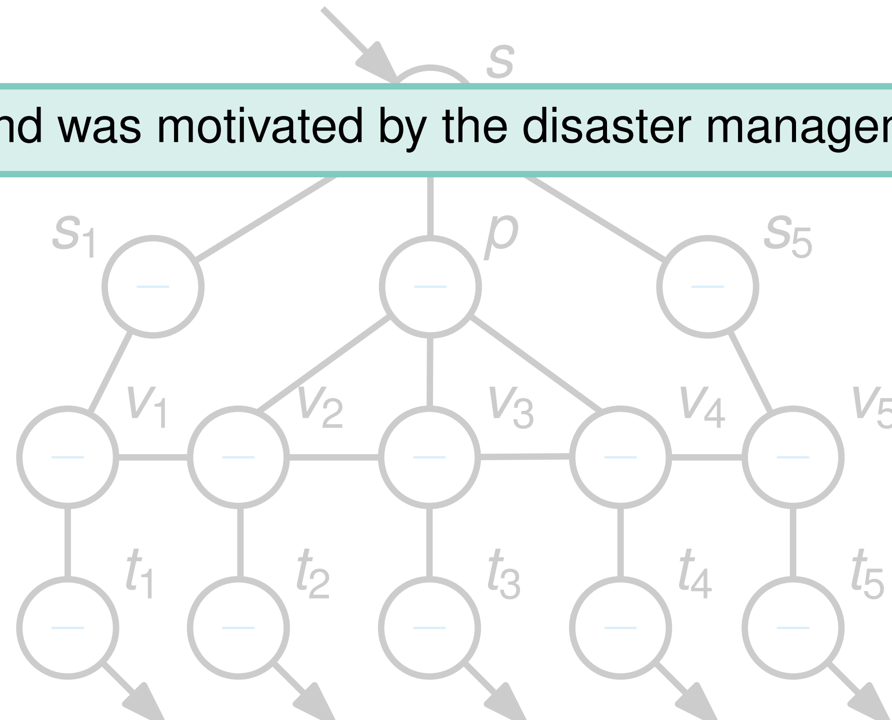
Definition 8 [page 8; Lehmann et al., 2014]

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 $t \in V_C$, 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



Institute of Theoretical Informatics
Algorithmics Group

Subset Sum Problem (SSP)

Decision Problem Subset Sum (SSP)

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

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

Institute of Theoretical Informatics
Algorithmics Group

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

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

Institute of Theoretical Informatics
Algorithmics Group

Subset Sum Problem (SSP)

Decision Problem Subset Sum (SSP)

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

Question: Is there a set of elements $x_1, x_2, \ldots, x_n \in \{0, 1\}$ 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.

Institute of Theoretical Informatics
Algorithmics Group

MTSF on 2-level Trees

[Lehmann et al., 2014]

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

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

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

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), $\forall i \in \{2, \dots, n\}$
  (v_i, t_i), $\forall i \in \{1, \dots, n+1\}$
  (v_{i-1}, v_i). $\forall i \in \{2, \dots, n+1\}$

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).

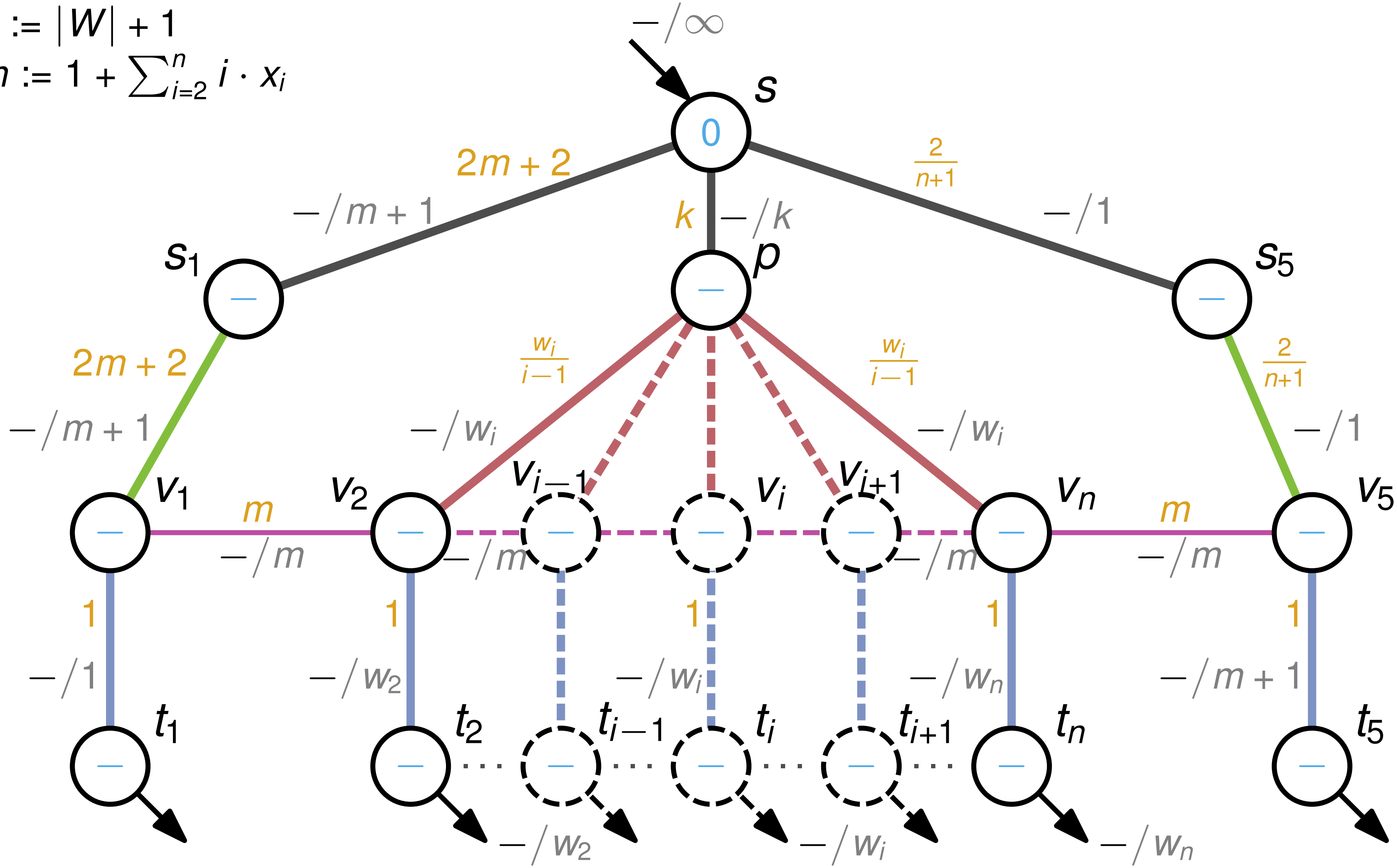
generation / load
- generation of $m + 2 + k$ at bus s,
  consumption of $m + 2 + k$ at buses t_i for all $1 \leq i \leq n + 1$

MTSF on 2-level Trees

[Lehmann et al., 2014]

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

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



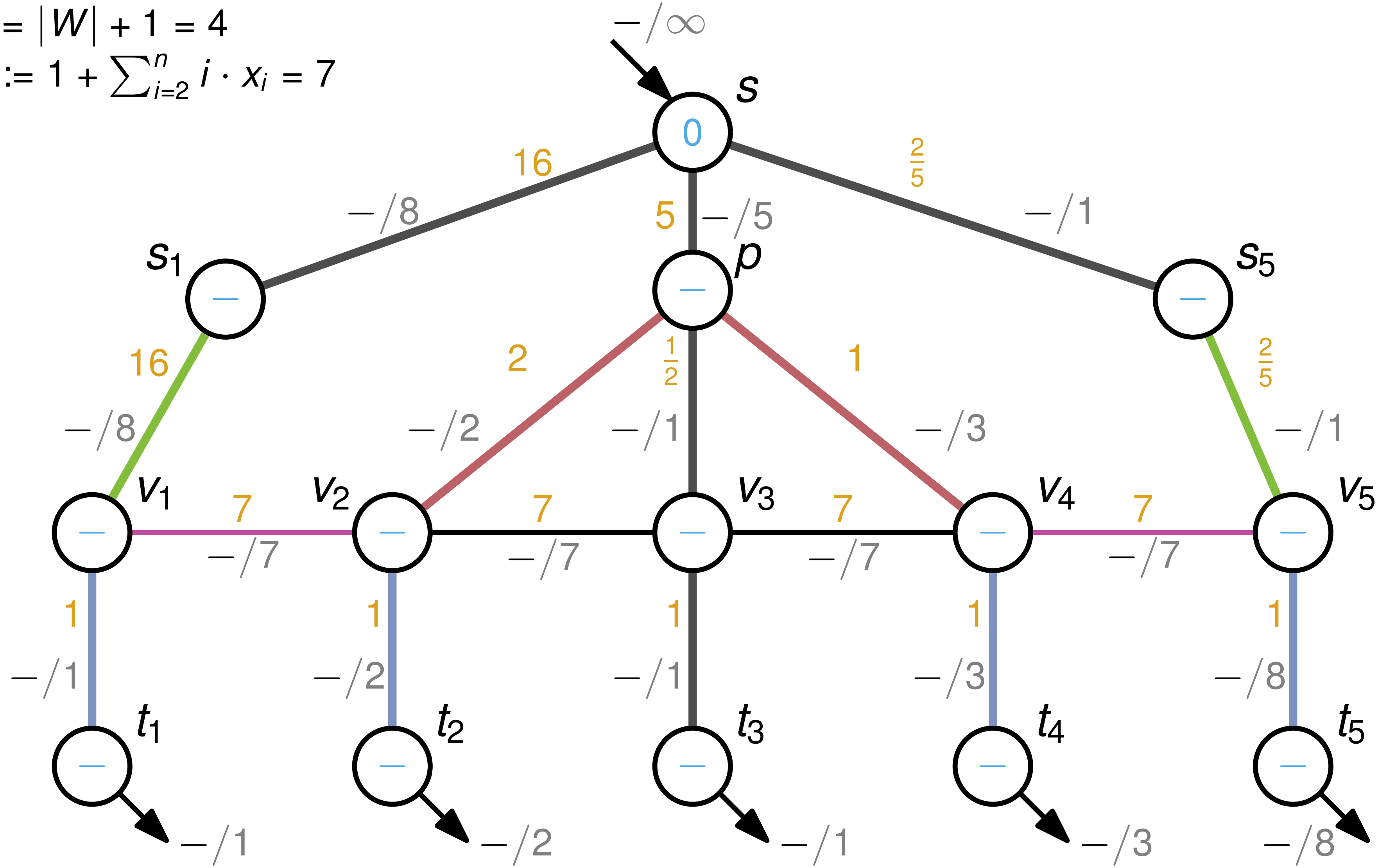
Institute of Theoretical Informatics
Algorithmics Group

MTSF on 2-level Trees

[Lehmann et al., 2014]

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

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

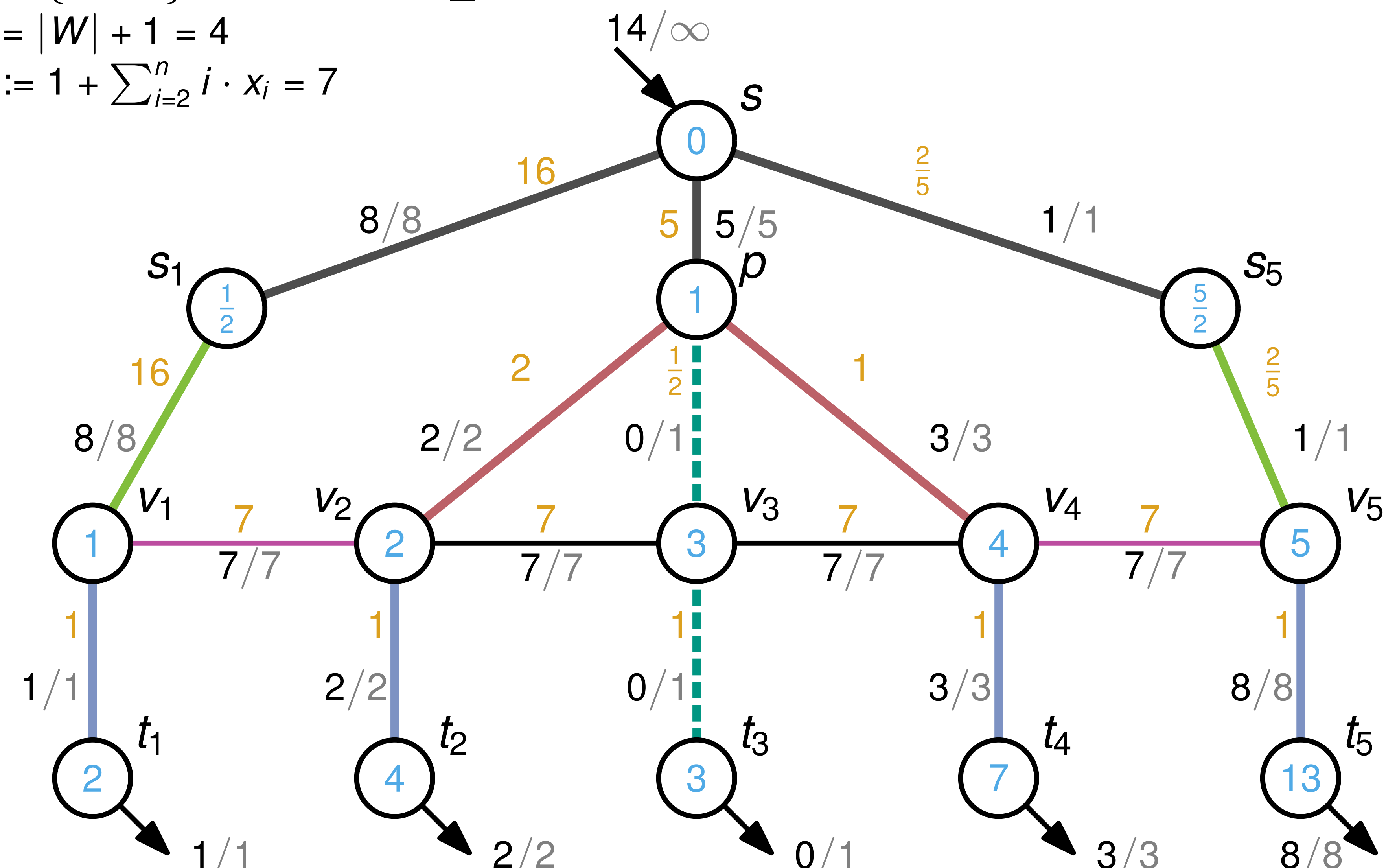


MTSF on 2-level Trees

[Lehmann et al., 2014]

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

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



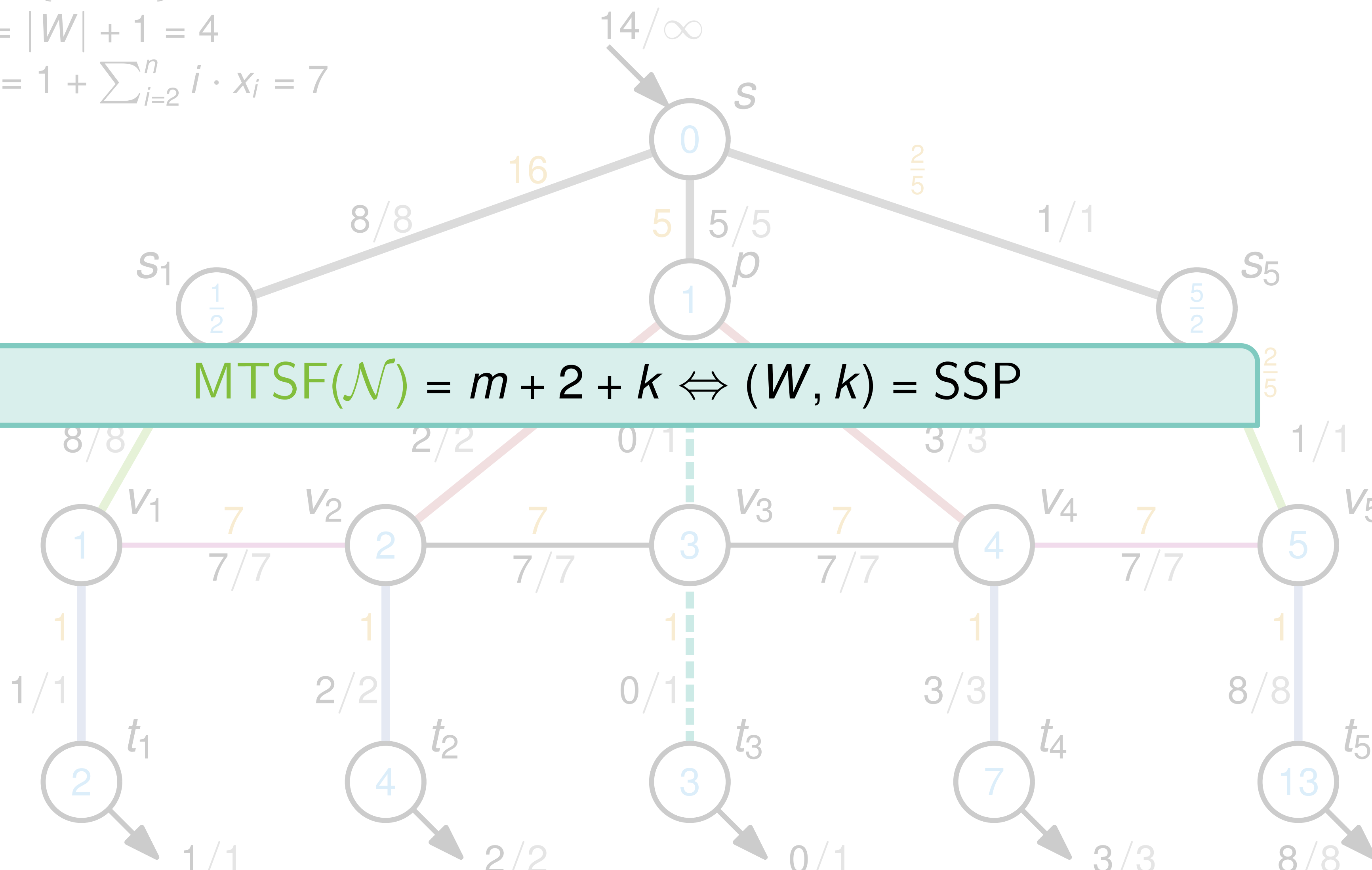
Institute of Theoretical Informatics
Algorithmics Group

MTSF on 2-level Trees

[Lehmann et al., 2014]

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

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

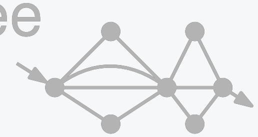
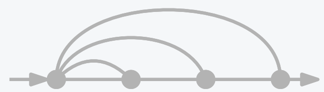
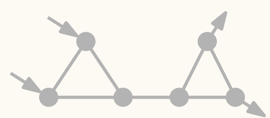
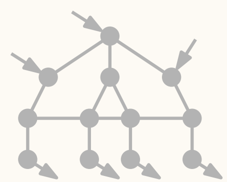
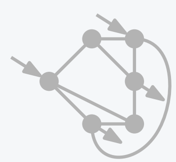
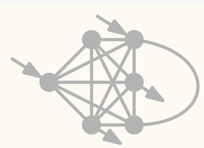


$$\text{MTSF}(\mathcal{N}) = m + 2 + k \Leftrightarrow (W, k) = \text{SSP}$$

Institute of Theoretical Informatics
Algorithmics Group

Overview of the MTSF Results

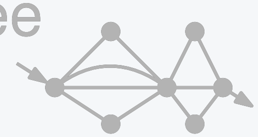
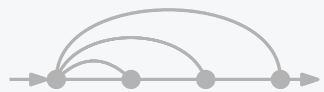
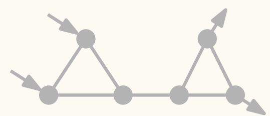
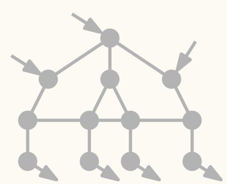
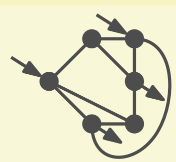
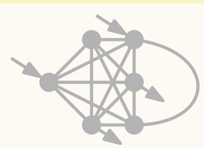
complexity

	Graph Structure		Complexity	Algorithm
one generator, one load	penrose-minor-free graphs		polynomial-time solvable	DTP ✓
	series-parallel graphs		NP-hard	✗
arbitrary generators, arbitrary loads	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]	✗
$\lvert V_G \rvert = 2$, $\lvert V_C \rvert = 2$	arbitrary graphs		non-APX [Lehmann et al., 2014]	✗

Institute of Theoretical Informatics
Algorithmics Group

Overview of the MTSF Results

complexity

	Graph Structure		Complexity	Algorithm
one generator, one load	penrose-minor-free graphs		polynomial-time solvable	DTP ✓
	series-parallel graphs		NP-hard	✗
arbitrary generators, arbitrary loads	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]	✗
$\|V_G\|=2$, $\|V_C\|=2$	arbitrary graphs		non-APX [Lehmann et al., 2014]	✗

Institute of Theoretical Informatics
Algorithmics Group

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 $\pi^\star \in \Pi$ 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]

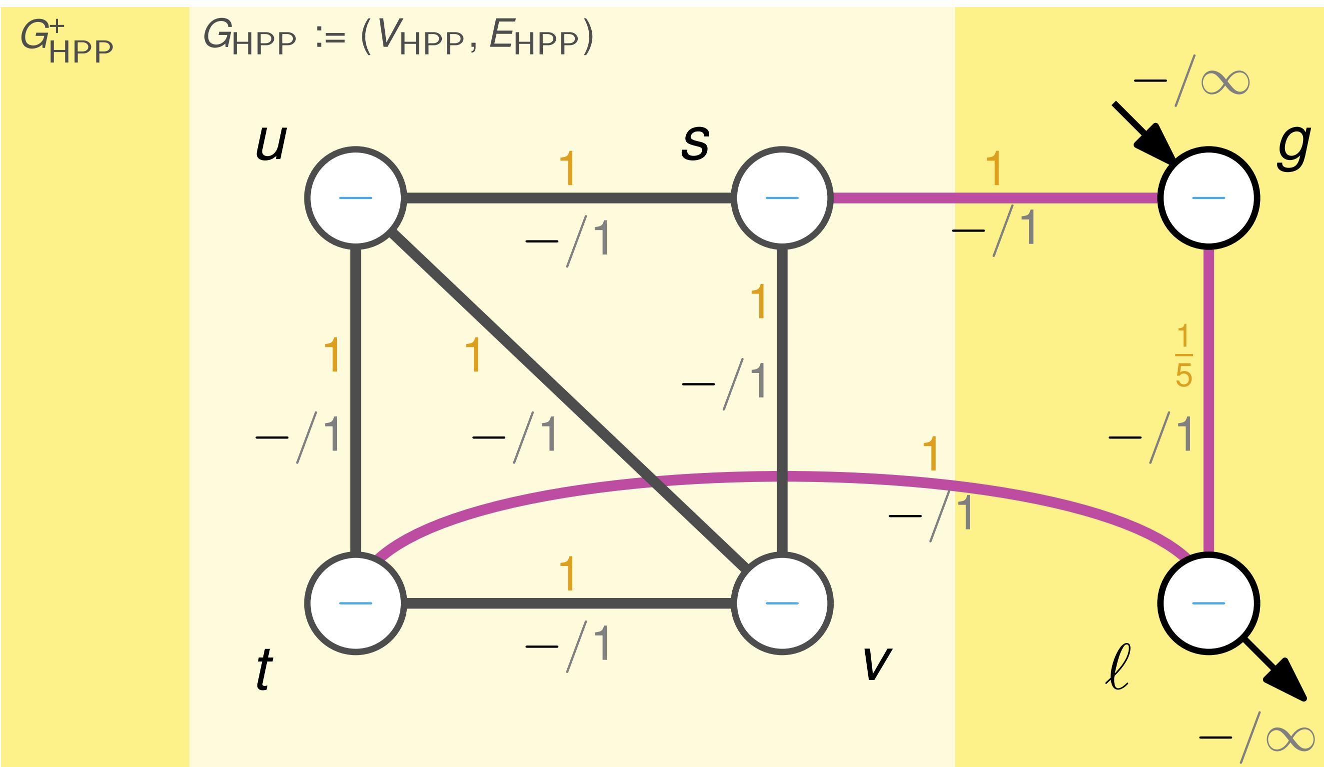
The feasibility problem for planar networks with a maximum degree of 3 is strongly NP-hard.

MTSF on Planar Graphs is strongly NP-hard

[Lehmann et al., 2014]

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

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



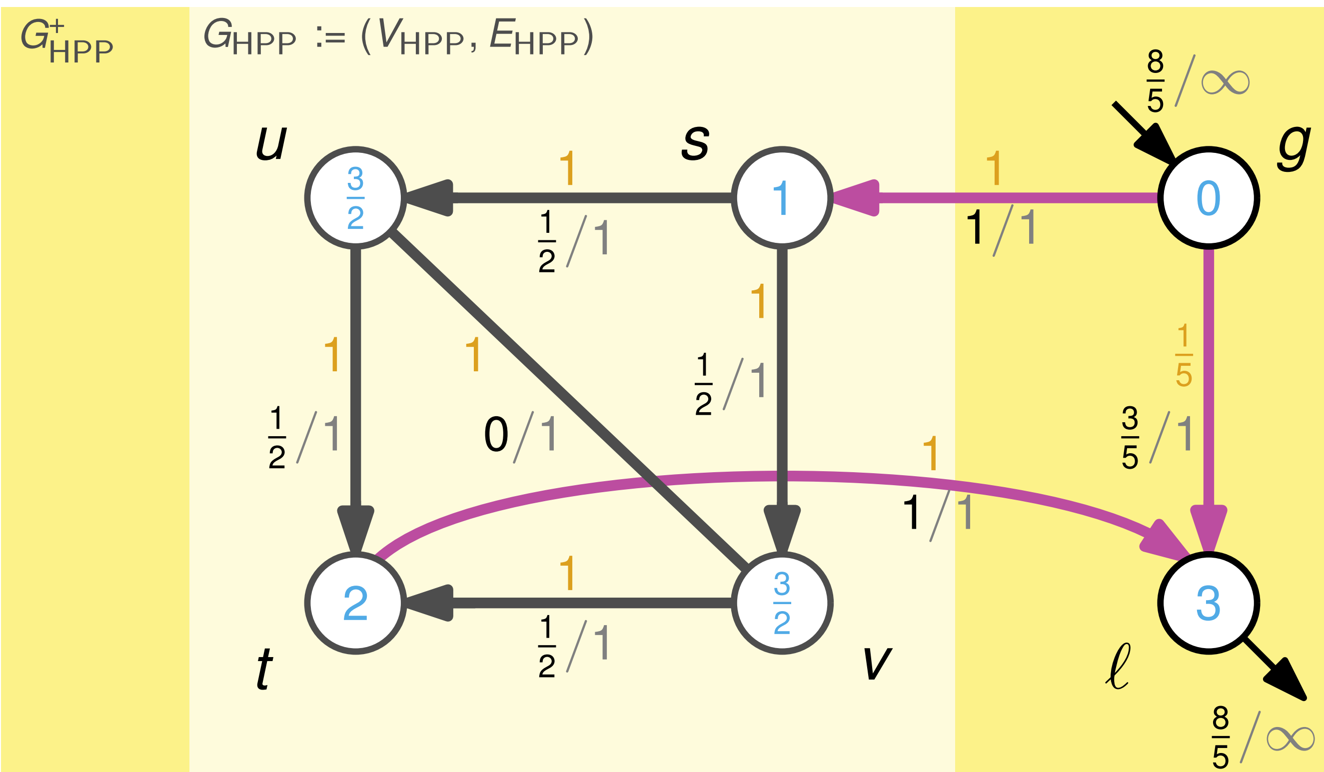
Institute of Theoretical Informatics
Algorithmics Group

MTSF on Planar Graphs is strongly NP-hard

[Lehmann et al., 2014]

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

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



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

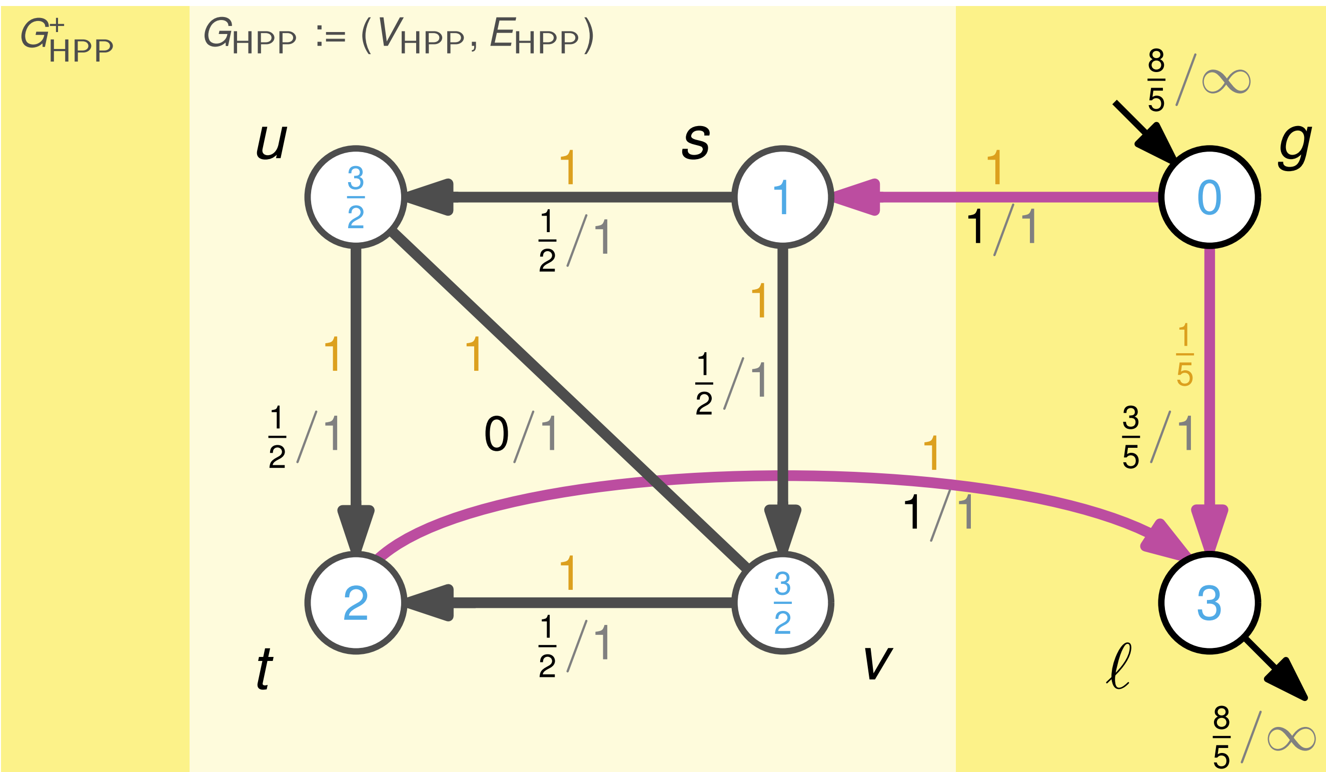
MTSF on Planar Graphs is strongly NP-hard

[Lehmann et al., 2014]

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

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

  $\text{cap}(g,\ell) := 1,\ b(g,\ell) := \frac{1}{n+1}$,



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

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

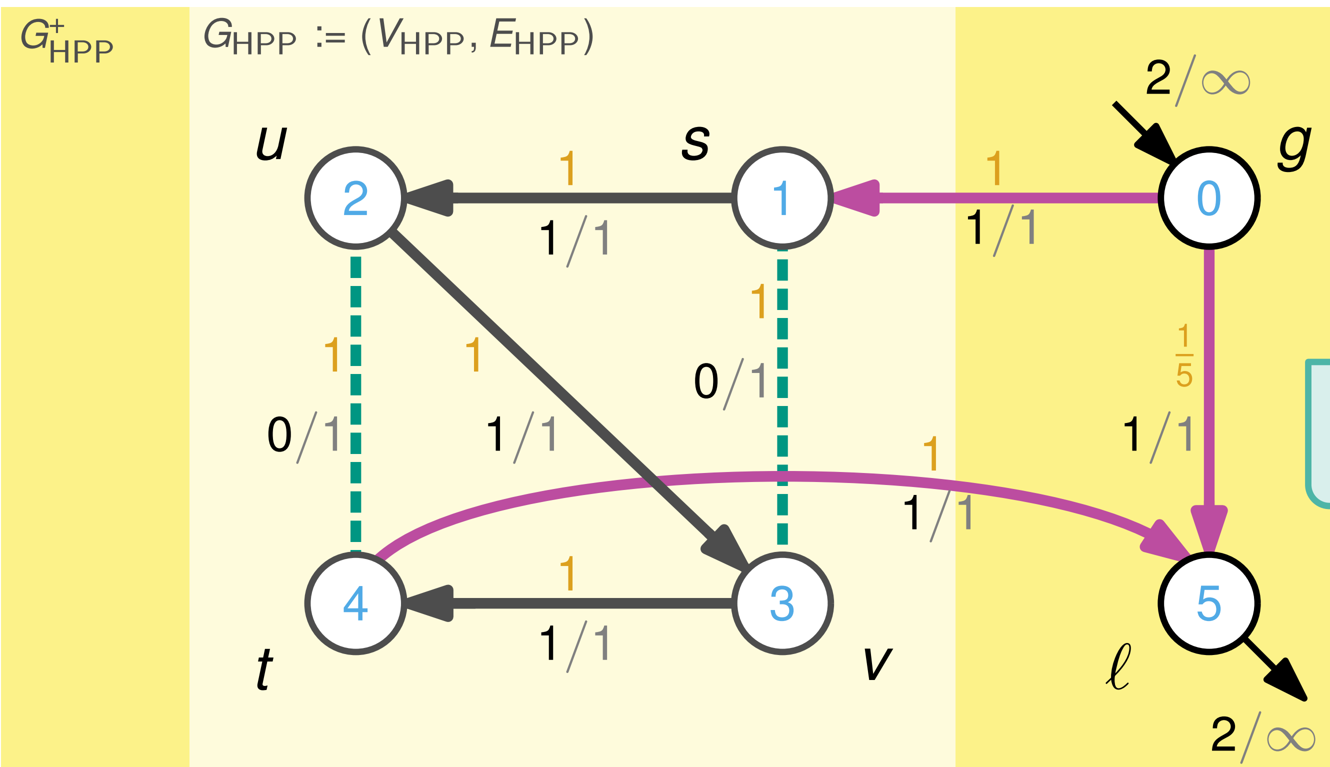
Institute of Theoretical Informatics
Algorithmics Group

$\color{green}{\text{MTSF}}$ on Planar Graphs is strongly NP-hard

[Lehmann et al., 2014]

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

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



OPT$_{\text{MTSF}}(\mathcal{N}) = 2$ iff G_{HPP} has a Hamilton Path. ✓

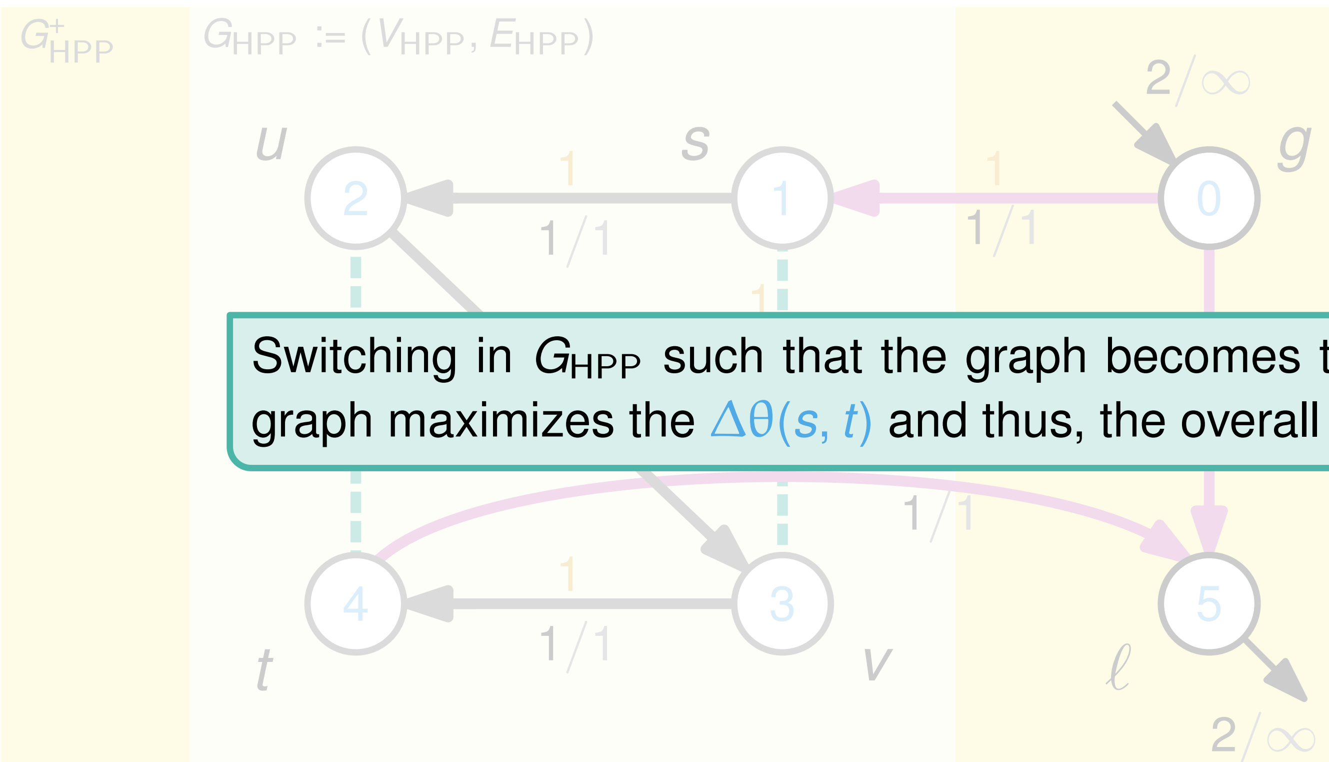
$\text{OPT}_{\text{MTSF}}(G^+_{\text{HPP}}) = 2$

MTSF on Planar Graphs is strongly NP-hard

[Lehmann et al., 2014]

Instance $\mathcal{N} = (G = (V \cup V_{\mathrm{HPP}}, E \cup E_{\mathrm{HPP}}), V_G = \{g\}, V_C = \{\ell\}, \mathrm{cap}, b, \overline{x}, \overline{d})$

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

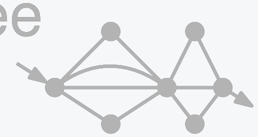
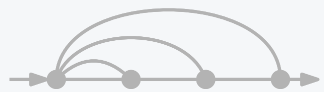
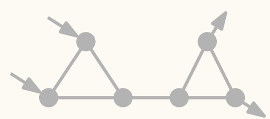
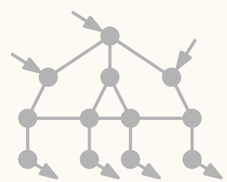
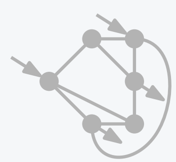
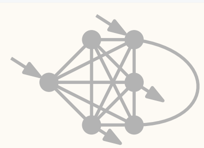


Switching in G_{HPP} such that the graph becomes the Hamiltonian graph maximizes the $\Delta\theta(s,t)$ and thus, the overall power flow.

$\mathrm{OPT}_{\mathrm{MTSF}}(G^+_{\mathrm{HPP}}) = 2$

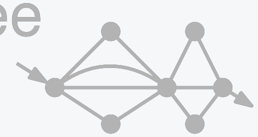
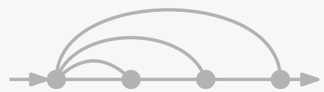
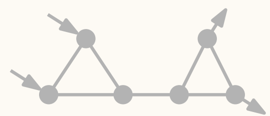
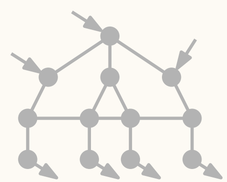
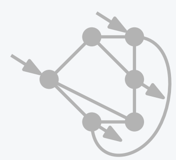
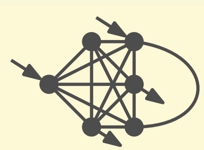
Overview of the MTSF Results

complexity

	Graph Structure	Complexity	Algorithm
one generator, one load	penrose-minor-free graphs 	polynomial-time solvable	DTP ✓
one generator, one load	series-parallel graphs 	NP-hard	✗
arbitrary generators, arbitrary loads	cacti with max degree of 3 	NP-hard [Lehmann et al., 2014]	2-approx. ✓
arbitrary generators, arbitrary loads	2-level trees 	NP-hard [Lehmann et al., 2014]	✗
	planar graphs with max degree of 3 	strongly NP-hard [Lehmann et al., 2014]	✗
$\lvert V_G \rvert = 2$, $\lvert V_C \rvert = 2$	arbitrary graphs 	non-APX [Lehmann et al., 2014]	✗

Overview of the MTSF Results

complexity

	Graph Structure		Complexity	Algorithm
one generator, one load	penrose-minor-free graphs		polynomial-time solvable	DTP ✓
	series-parallel graphs		NP-hard	✗
arbitrary generators, arbitrary loads	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]	✗
$\lvert V_G \rvert = 2$, $\lvert V_C \rvert = 2$	arbitrary graphs		non-APX [Lehmann et al., 2014]	✗

Institute of Theoretical Informatics
Algorithmics Group

LONGEST PATH PROBLEM (LPP)

[Karger et al., 1997]

LONGEST PATH PROBLEM (LPP)

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

Question: Find a simple path $\pi^\star \in \Pi(s,t)$ 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 $\mathcal{N}_{\text{LPP}}$.

- Also called TAXICAB RIPOFF problem

Longest Path Problem (LPP)

[Karger et al., 1997]

Longest Path Problem (LPP)

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

Question: Find a simple path $\pi^\star \in \Pi(s,t)$ 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 $\mathcal{N}_{\text{LPP}}$.

- Also called TAXICAB RIPOFF problem
- For all $\epsilon > 0$ not possible to approximate LPP 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]

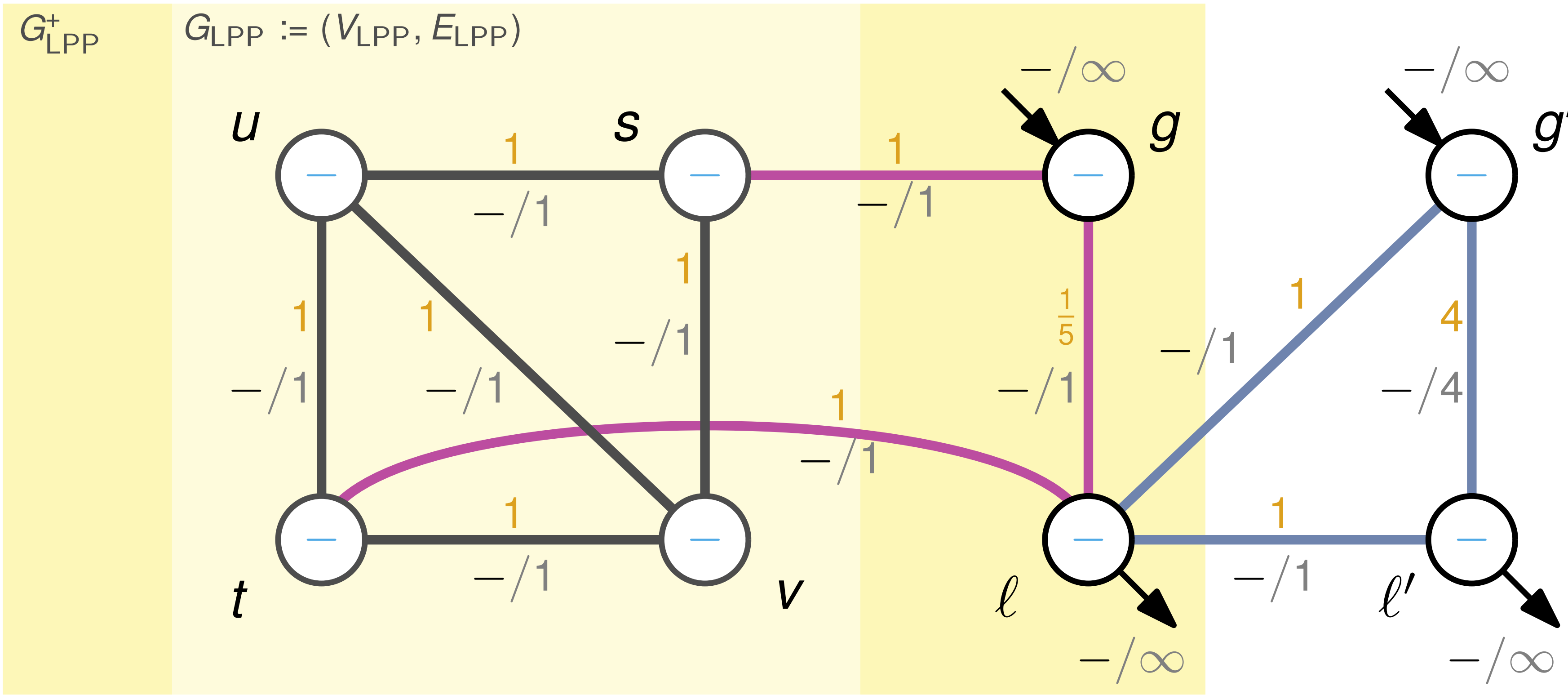
It is not possible to approximate the MTSF problem within a factor of $2^{(\log n)^{1-\epsilon}}$ unless NP is contained within quasi-polynomial deterministic time.

MTSF on Arbitrary Graphs is non-APX

[Lehmann et al., 2014]

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

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



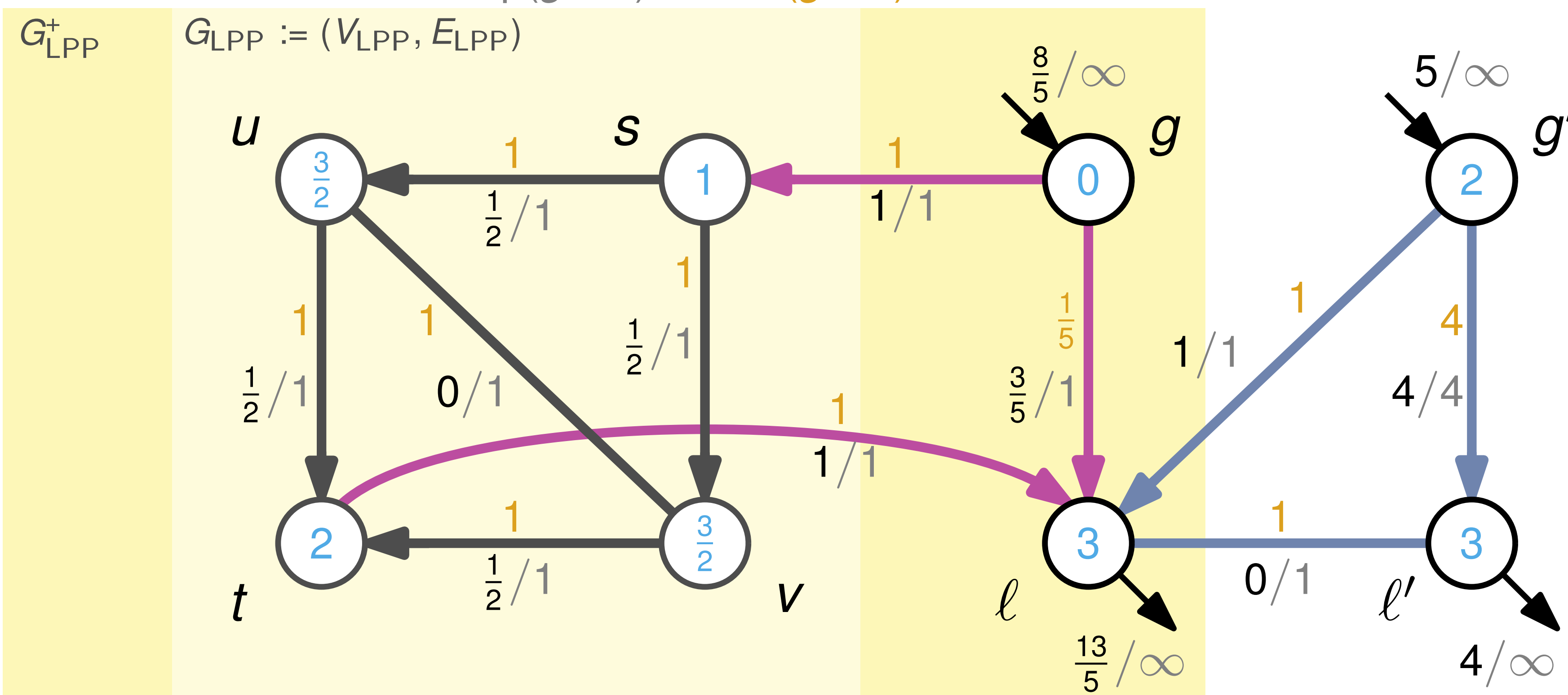
Institute of Theoretical Informatics
Algorithmics Group

MTSF on Arbitrary Graphs is non-APX

[Lehmann et al., 2014]

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

- Set of vertices $V_{\text{LPP}} = \{v_1, v_2, \ldots, v_n\}$, and $\overline{x} = \overline{d} = \infty$,
- Set of edges $E = E_{\text{LPP}} \cup \{(g, s), (t, \ell), (g, \ell), (g', \ell'), (g', \ell), (\ell', \ell)\}$,
- 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

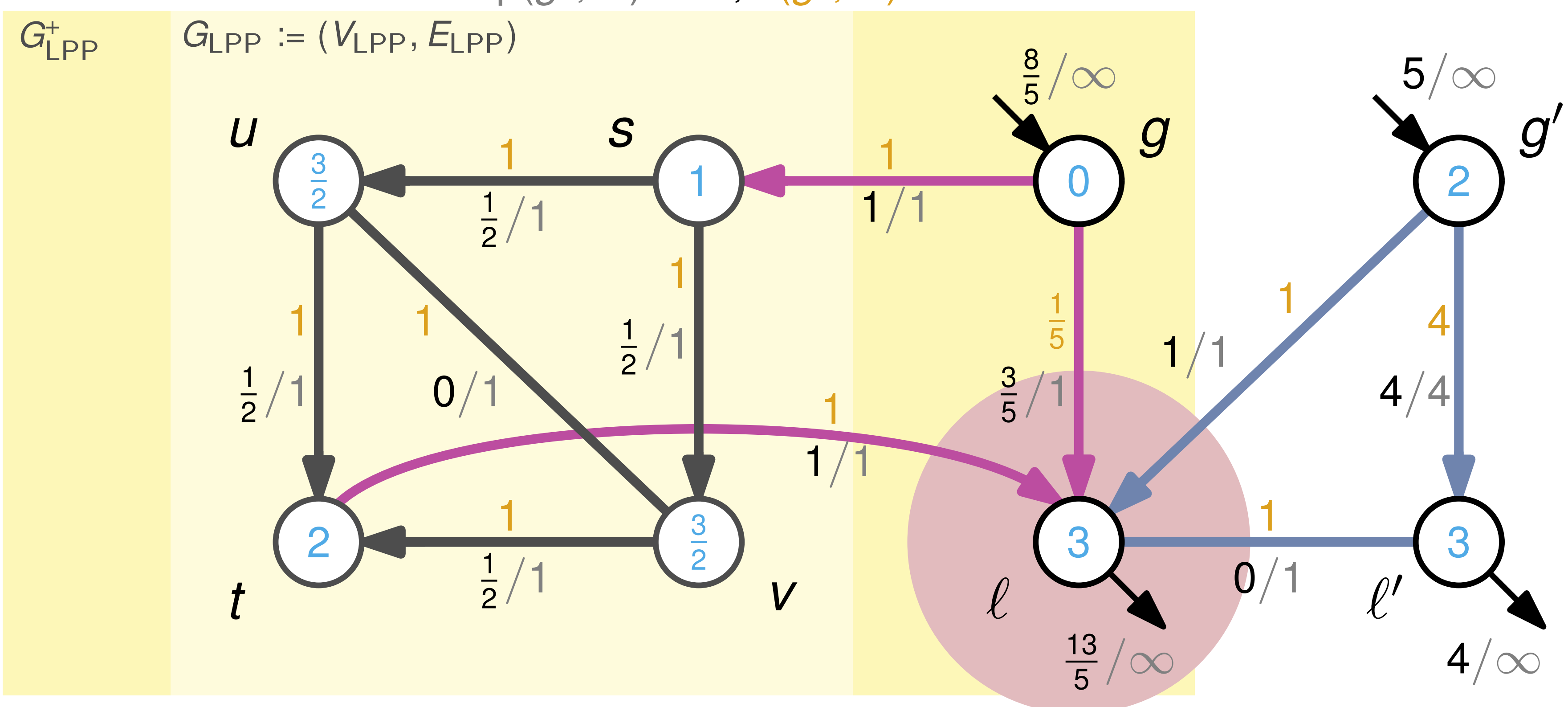
[Lehmann et al., 2014]

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

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

  $\text{cap}(g, \ell) := 1,\ b(g, \ell) := \frac{1}{n+1}$,

  $\text{cap}(g', \ell') := n,\ b(g', \ell') := n$



$3 \leq \frac{13}{5} \leq \infty$ ↯

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

MTSF on Arbitrary Graphs is non-APX

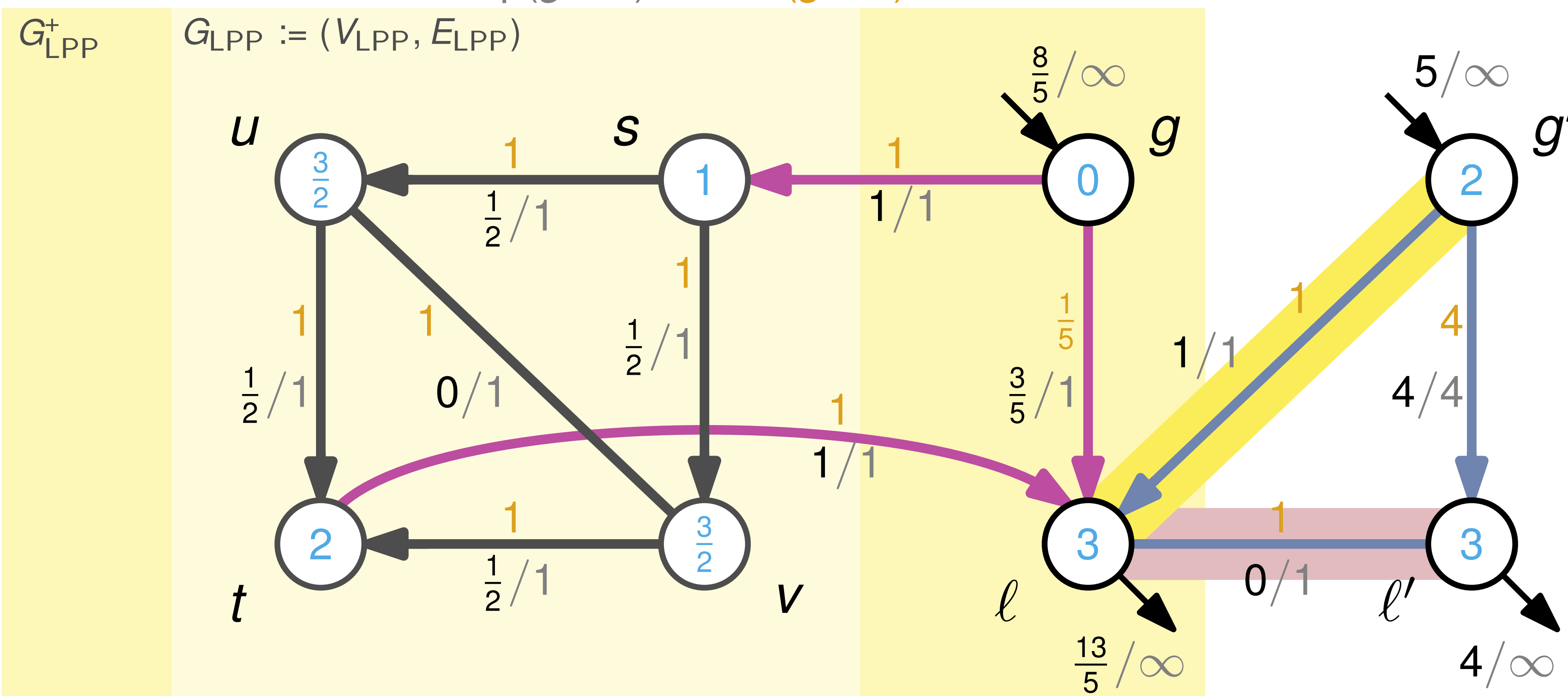
[Lehmann et al., 2014]

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

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

  $\text{cap}(g,\ell) := 1, b(g,\ell) := \frac{1}{n+1}$,

  $\text{cap}(g',\ell') := n, b(g',\ell') := n$



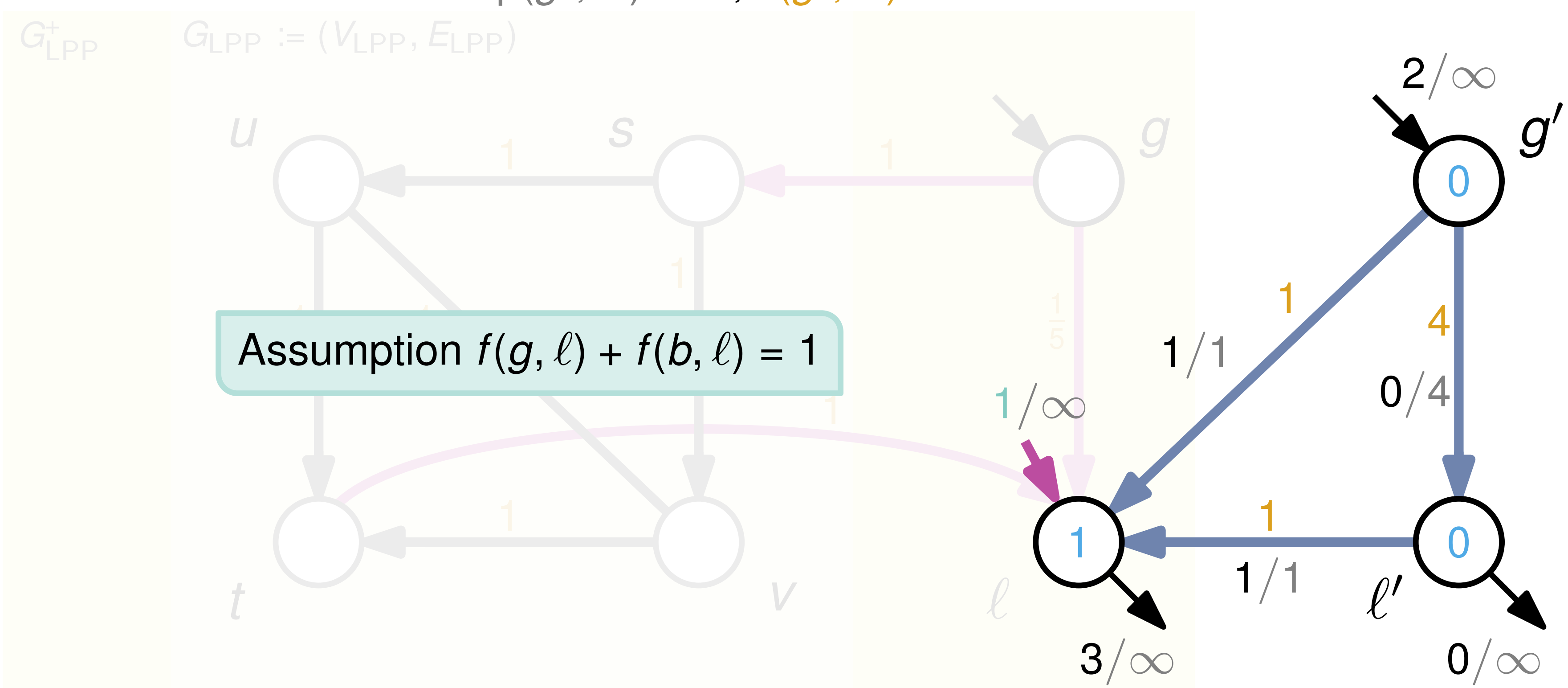
$\text{OPT}_{\text{MPF}}(\mathcal{N}) = \frac{33}{5}$

MTSF on Arbitrary Graphs is non-APX

[Lehmann et al., 2014]

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

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



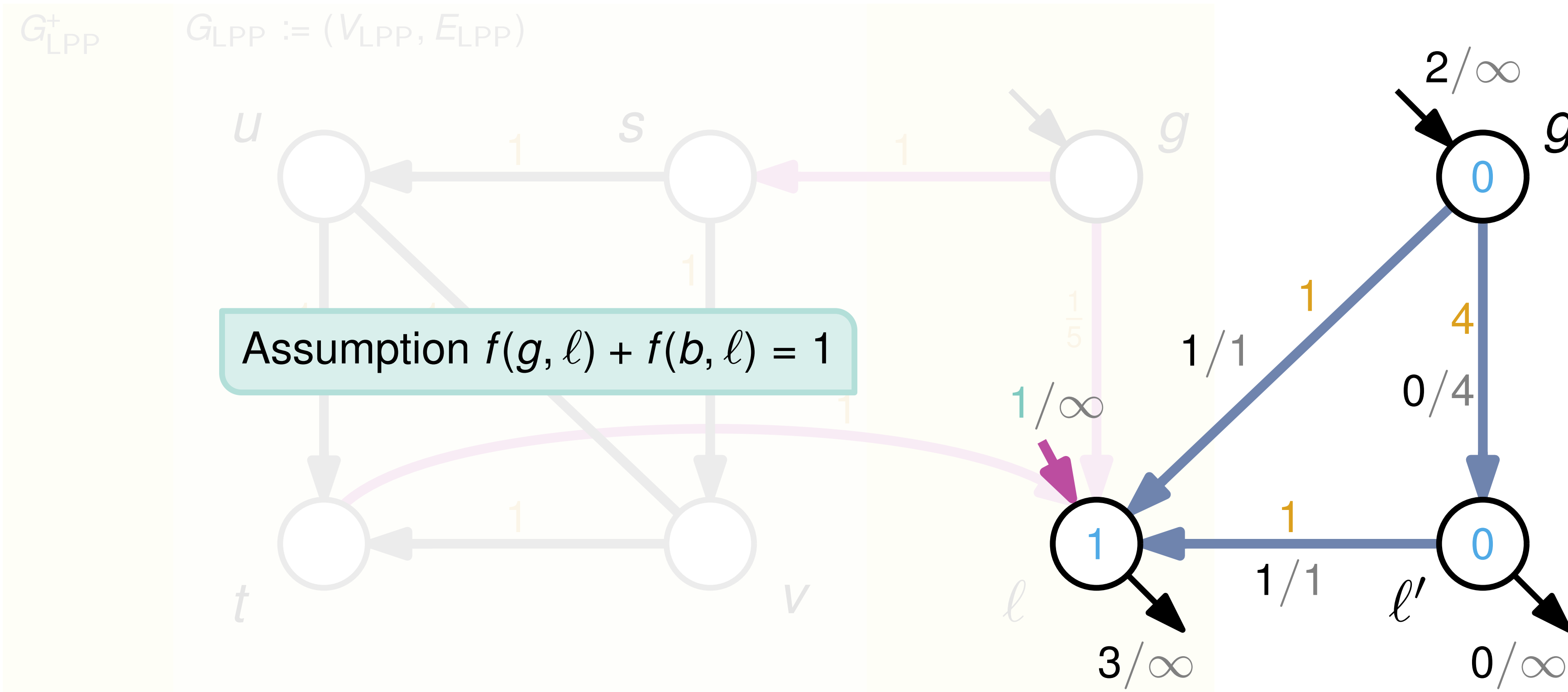
Institute of Theoretical Informatics
Algorithmics Group

MTSF on Arbitrary Graphs is non-APX

[Lehmann et al., 2014]

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

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



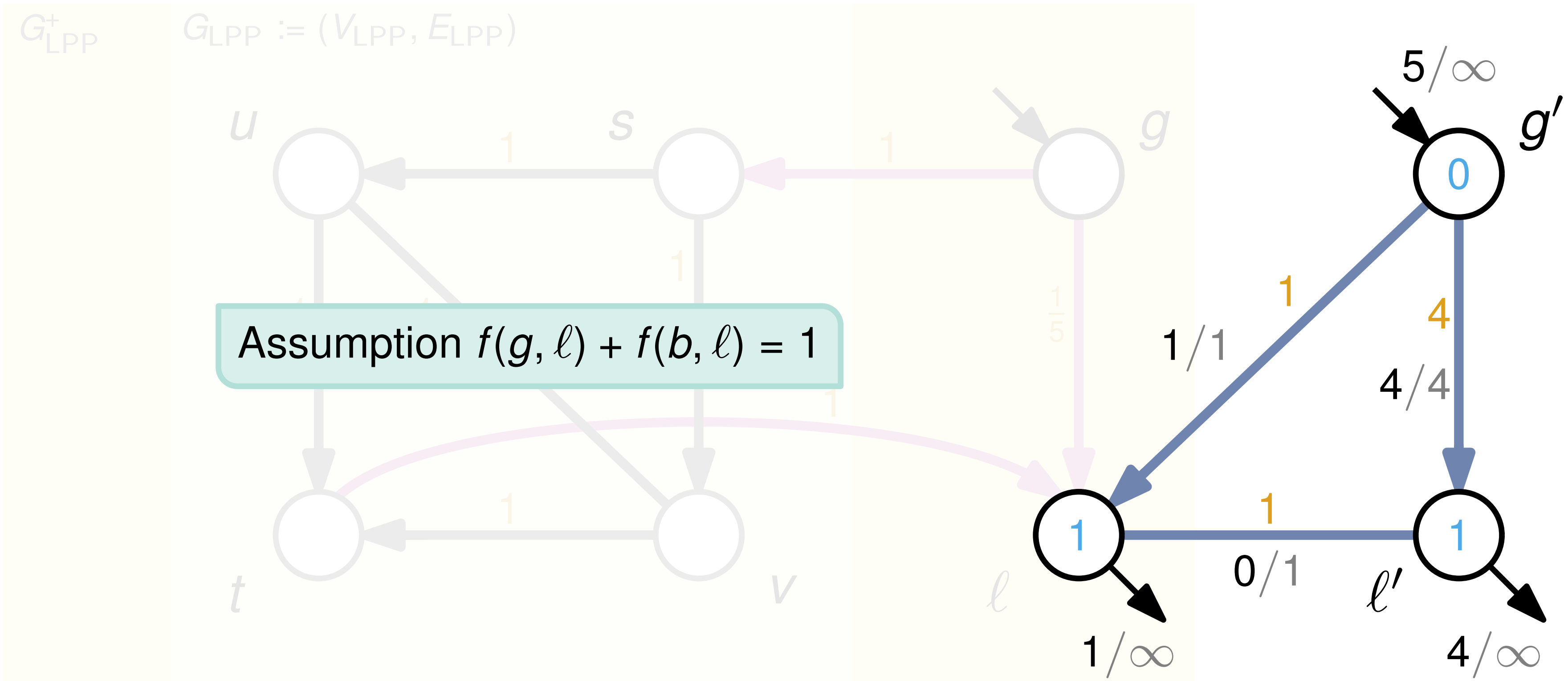
What is the maximum possible power flow in $(\mathcal{N} \setminus G_{\text{LPP}}) - g$?

MTSF on Arbitrary Graphs is non-APX

[Lehmann et al., 2014]

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

- Set of vertices $V_{\text{LPP}} = \{v_1, v_2, \ldots, v_n\}$, and $\overline{x} = \overline{d} = \infty$,
- Set of edges $E = E_{\text{LPP}} \cup \{(g, s), (t, \ell), (g, \ell), (g', \ell'), (g', \ell), (\ell', \ell)\}$,
- 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 $(\mathcal{N} \setminus G_{\text{LPP}}) - g$?

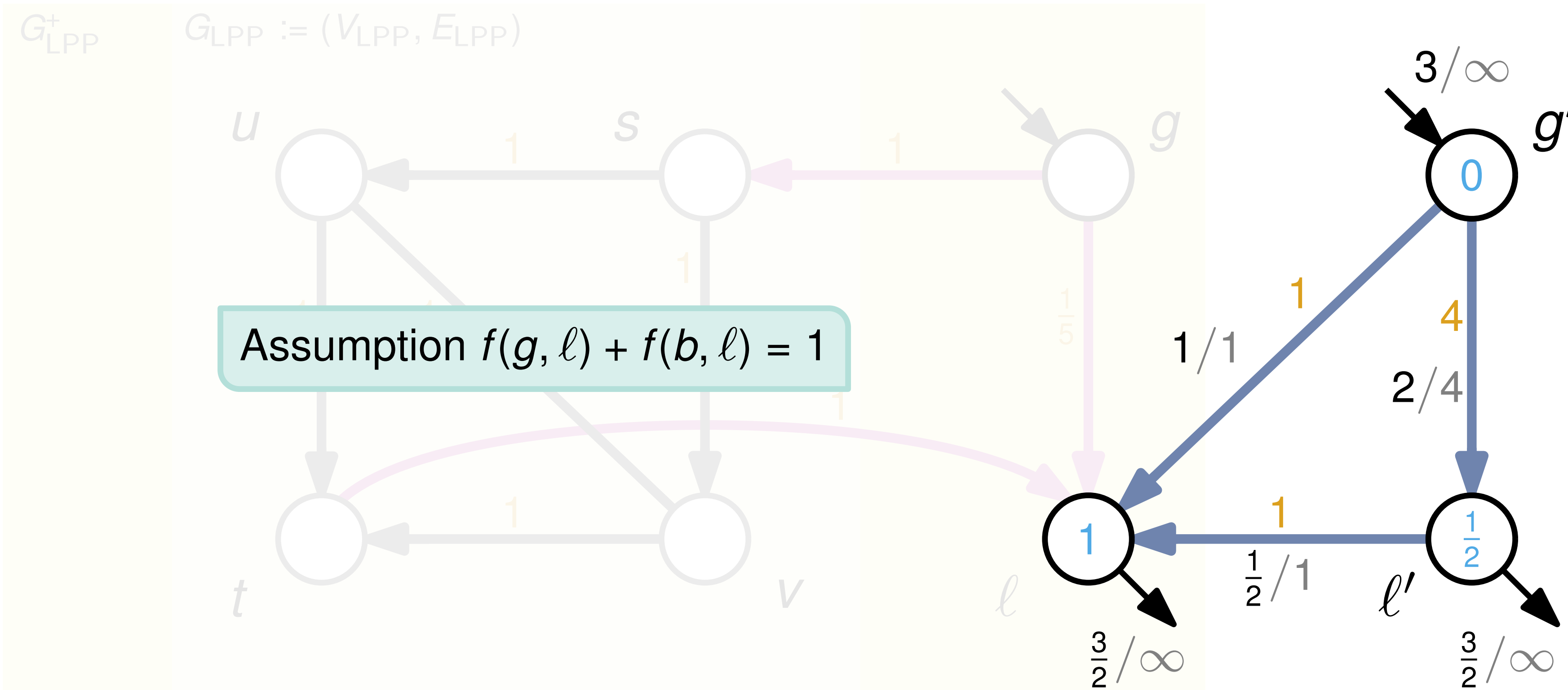
$\text{OPT}_{\text{MPF}}((\mathcal{N} \setminus G_{\text{LPP}}) - g) = 5$

MTSF on Arbitrary Graphs is non-APX

[Lehmann et al., 2014]

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

- Set of vertices $V_{\text{LPP}} = \{v_1, v_2, \ldots, v_n\}$, and $\overline{x} = \overline{d} = \infty$,
- Set of edges $E = E_{\text{LPP}} \cup \{(g, s), (t, \ell), (g, \ell), (g', \ell'), (g', \ell), (\ell', \ell)\}$,
- 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 $(\mathcal{N} \setminus G_{\text{LPP}}) - g$? — $\max f_{\text{net}}(\ell) = 3$

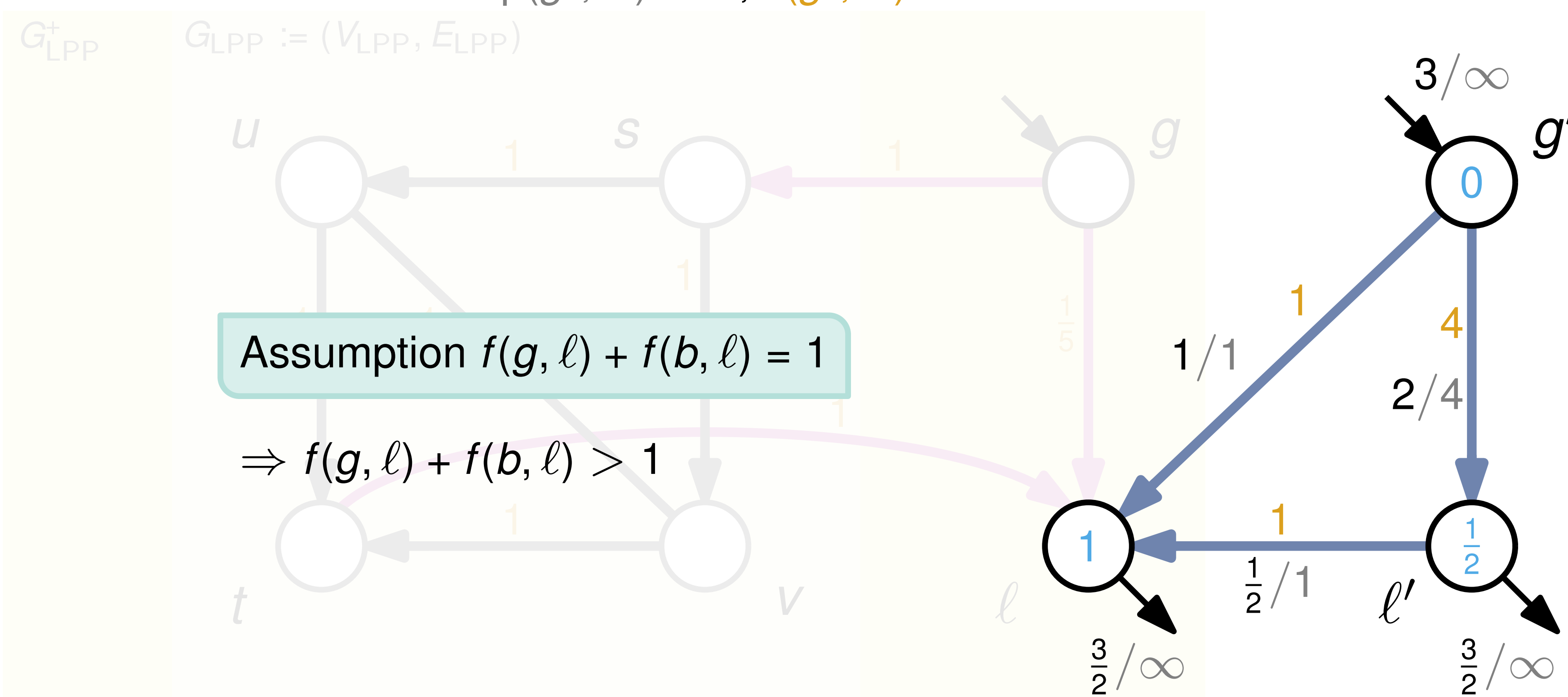
Institute of Theoretical Informatics
Algorithmics Group

MTSF on Arbitrary Graphs is non-APX

[Lehmann et al., 2014]

Instance $\mathcal{N} = (G = (V \cup V_{\mathrm{LPP}}, E \cup E_{\mathrm{LPP}}), V_G = \{g, g'\}, V_C = \{\ell, \ell'\}, \mathrm{cap}, b, \overline{x}, \underline{d}_\ell = 3, \overline{d})$

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



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

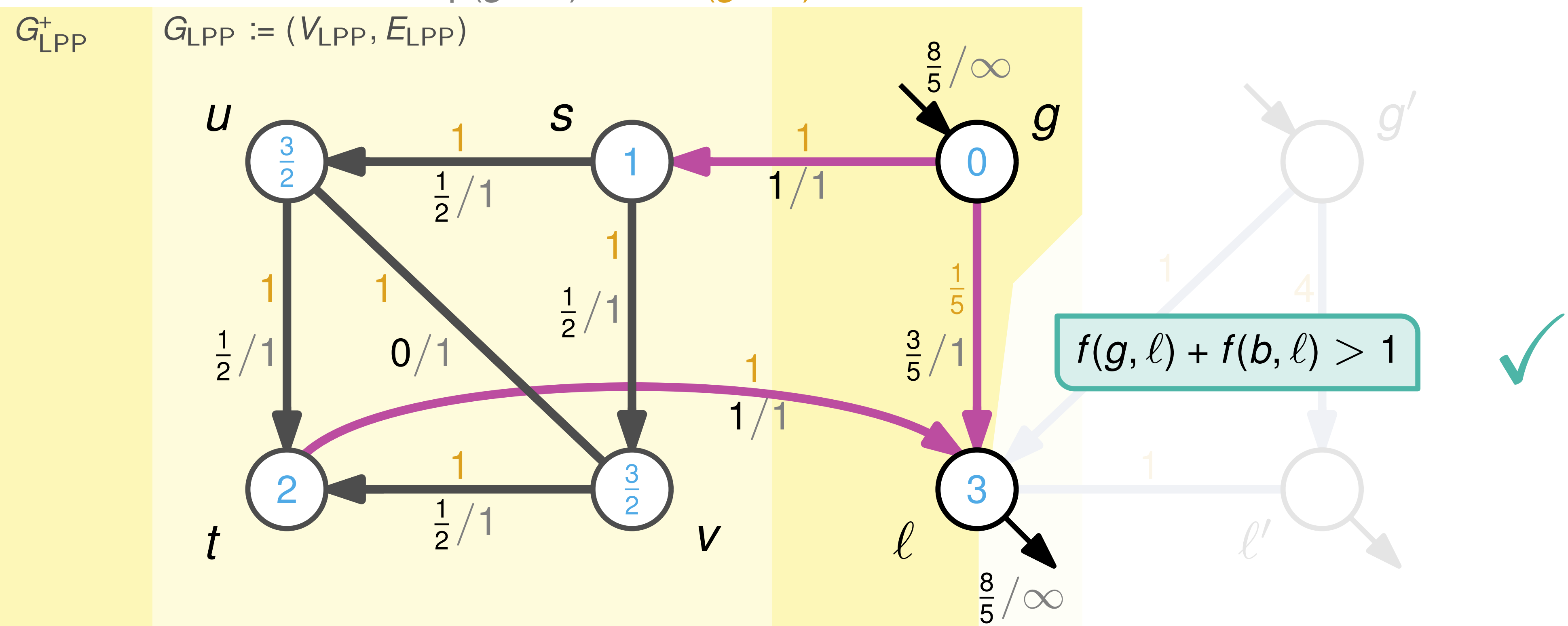
— $\max f_{\mathrm{net}}(\ell) = 3$

MTSF on Arbitrary Graphs is non-APX

[Lehmann et al., 2014]

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

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



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

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

MTSF on Arbitrary Graphs is non-APX

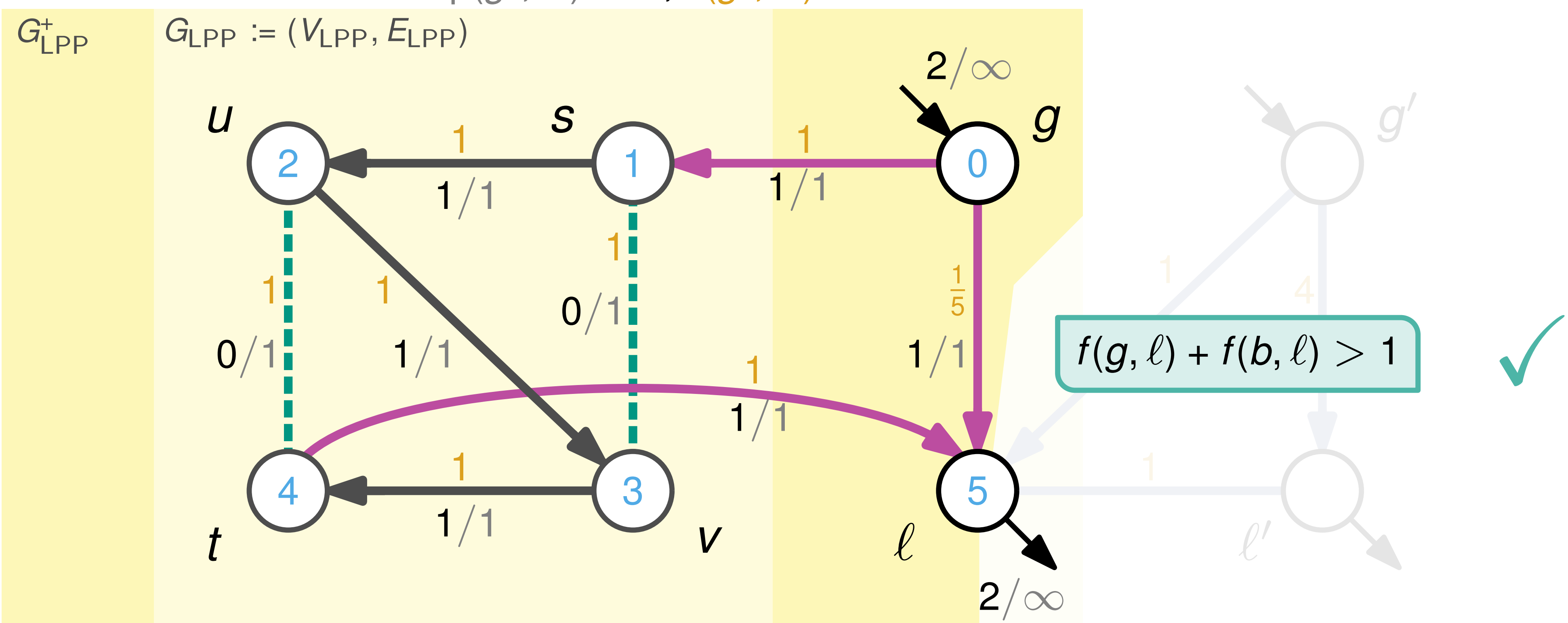
[Lehmann et al., 2014]

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

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

$\text{cap}(g, \ell) := 1, b(g, \ell) := \frac{1}{n+1}$,

$\text{cap}(g', \ell') := n, b(g', \ell') := n$



$\text{OPT}_{\text{MTSF}}(G^+_{LPP}) = 2$

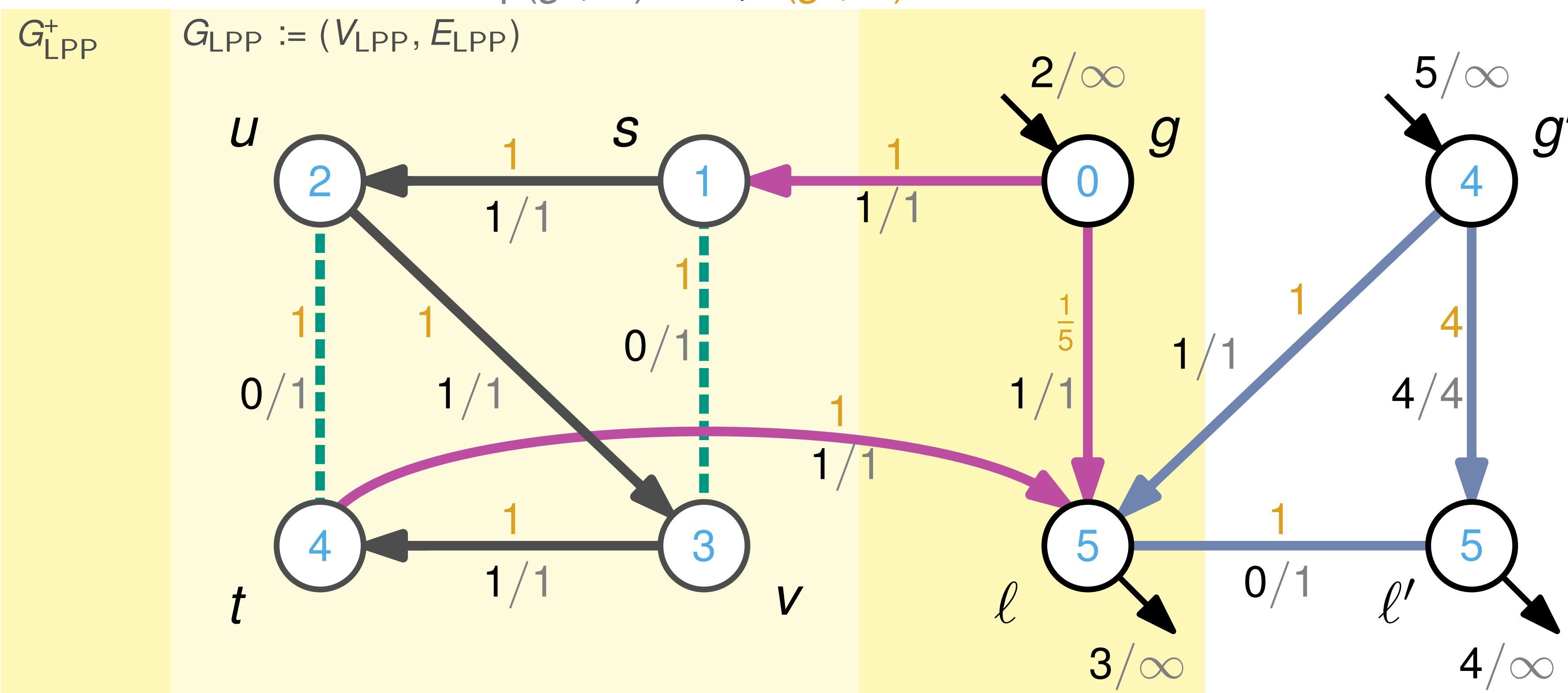
Institute of Theoretical Informatics
Algorithmics Group

MTSF on Arbitrary Graphs is non-APX

[Lehmann et al., 2014]

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

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



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

Institute of Theoretical Informatics
Algorithmics Group

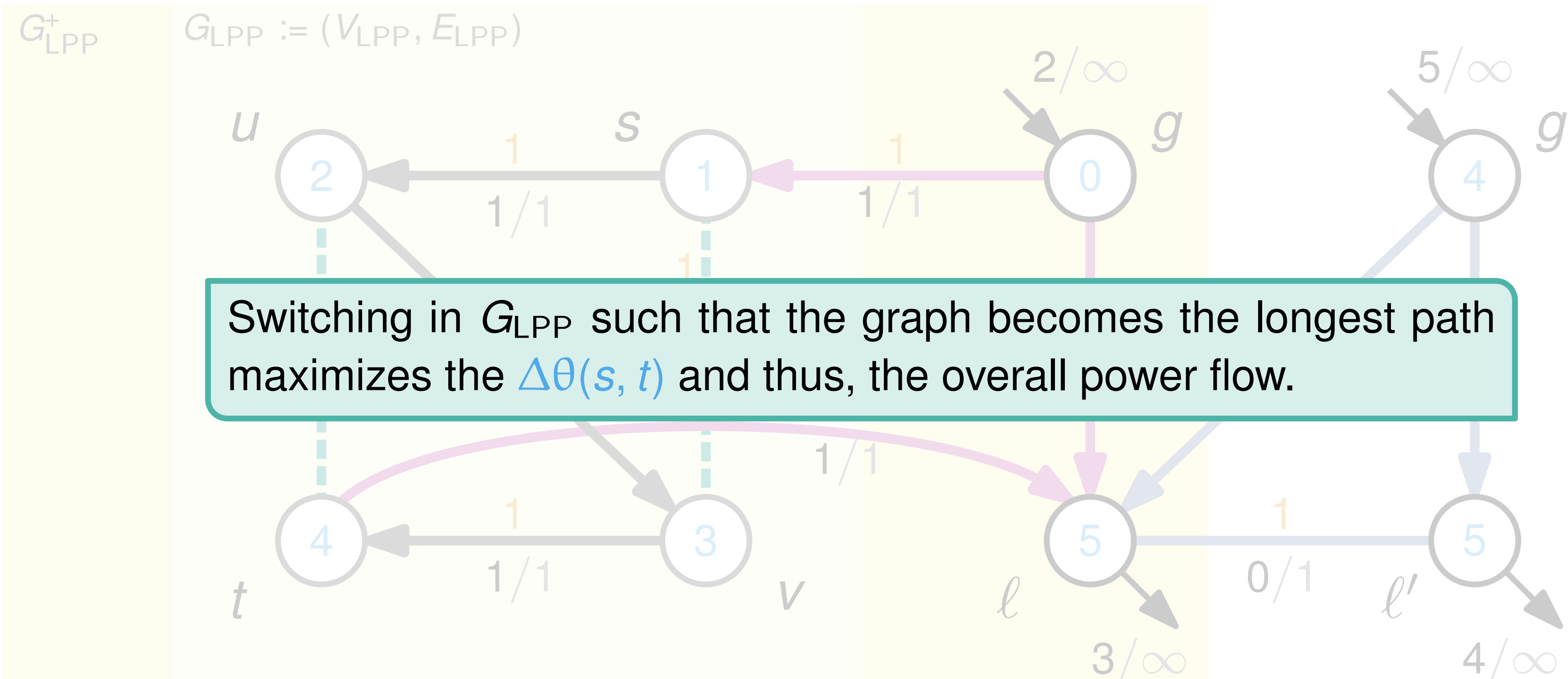
$\textbf{MTSF}$ on Arbitrary Graphs is non-APX

[Lehmann et al., 2014]

Karlsruhe Institute of Technology

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

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



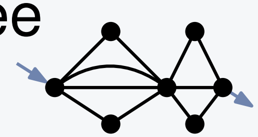
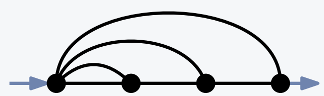
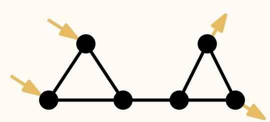
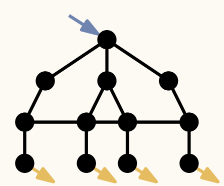
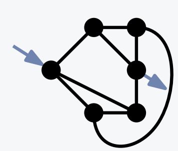
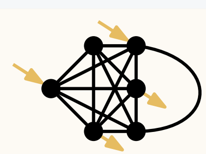
Switching in G_{LPP} such that the graph becomes the longest path maximizes the $\Delta\theta(s, t)$ and thus, the overall power flow.

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

Summary & Future Work

Karlsruhe Institute of Technology

complexity

	Graph Structure		Complexity	Algorithm
one generator, one load	penrose-minor-free graphs		polynomial-time solvable	✓
one generator, one load	series-parallel graphs		NP-hard	✗
arbitrary generators, arbitrary loads	cacti with max degree of 3		NP-hard [Lehmann et al., 2014]	✓
arbitrary generators, arbitrary loads	2-level trees		NP-hard [Lehmann et al., 2014]	✗
	planar graphs with max degree of 3		strongly NP-hard [Lehmann et al., 2014]	✗
$\lvert V_G\rvert=2$, $\lvert V_C\rvert=2$	arbitrary graphs		non-APX [Lehmann et al., 2014]	✗

Institute of Theoretical Informatics
Algorithmics Group

Summary & Future Work

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

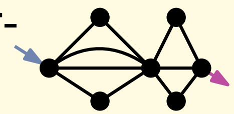
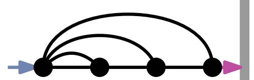
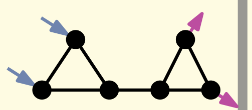
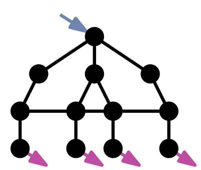
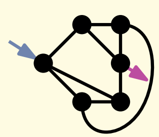
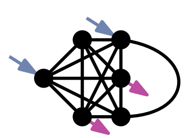
- What happens if we minimize the number of switches or fix a set of non-switchable edges?
- Is there a PTAS on cacti for MTSF?
- Replace ✗ by ✓

Institute of Theoretical Informatics
Algorithmics Group

Results

NESTA Case	n	m	$\lvert S_{\mathrm{MTSF}}\rvert$	$\lvert S_{\mathrm{MaxST}}\rvert$	$\mathrm{OPT}_{\mathrm{PF}}$	$\mathrm{OPT}_{\mathrm{MPF}}$	$\mathrm{OPT}_{\mathrm{MTSF}}$	$\mathrm{OPT}_{\mathrm{MaxST}}$	$\mathrm{OPT}_{\mathrm{MF}}$	max Gen
3_lmbd	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	$\lvert V_G \rvert$	$\lvert V_C \rvert$	b	cap	Complexity	Algorithms
MTSF and OTS	penrose-minor-free graphs 	1	1	—	—	polynomial-time solvable [Grastien et al., 2018]	✓ better? ∞ ∞
MTSF and OTS	series-parallel graphs 	1	1	∞ ∞	∞ 1	[Kocuk et al., 2016] NP-hard [Grastien et al., 2018]	✗
MTSF and OTS	cacti with max degree of 3 	∞	∞	1	∞	NP-hard [Lehmann et al., 2014]	✓ better? ∞ ∞
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 1	2 ∞	∞	∞	non-APX [Lehmann et al., 2014]	✗

Other interesting structures?

Stronger results?

Any algorithm giving guarantees?

Institute of Theoretical Informatics
Algorithmics Group

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.