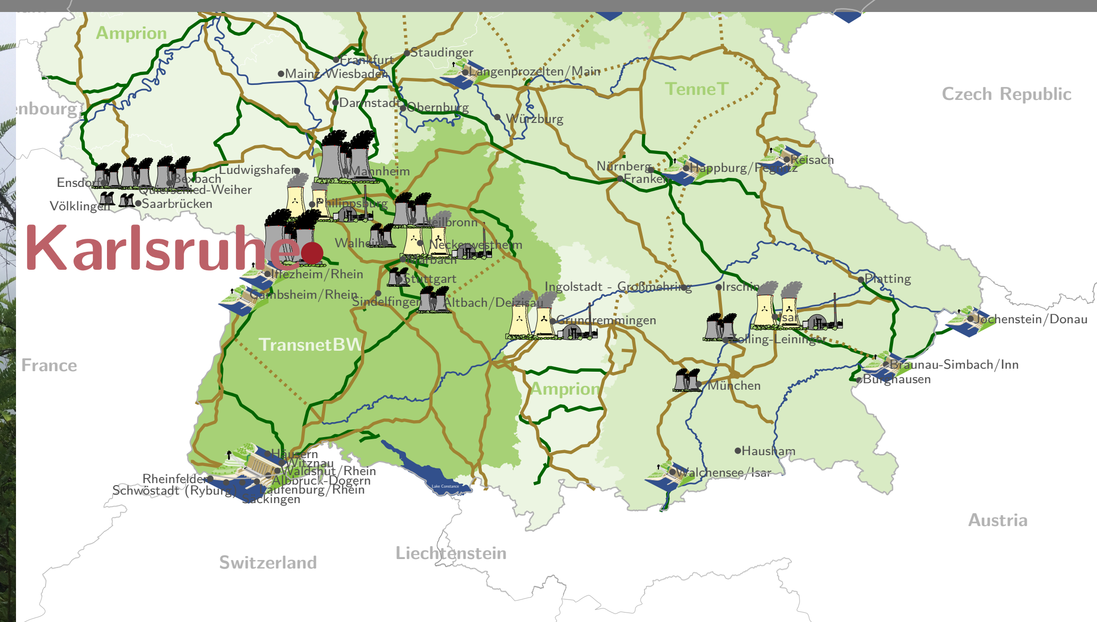
Karlsruher Institut für Technologie

Das Maximale Schaltungsflussproblem für Übertragungsnetze

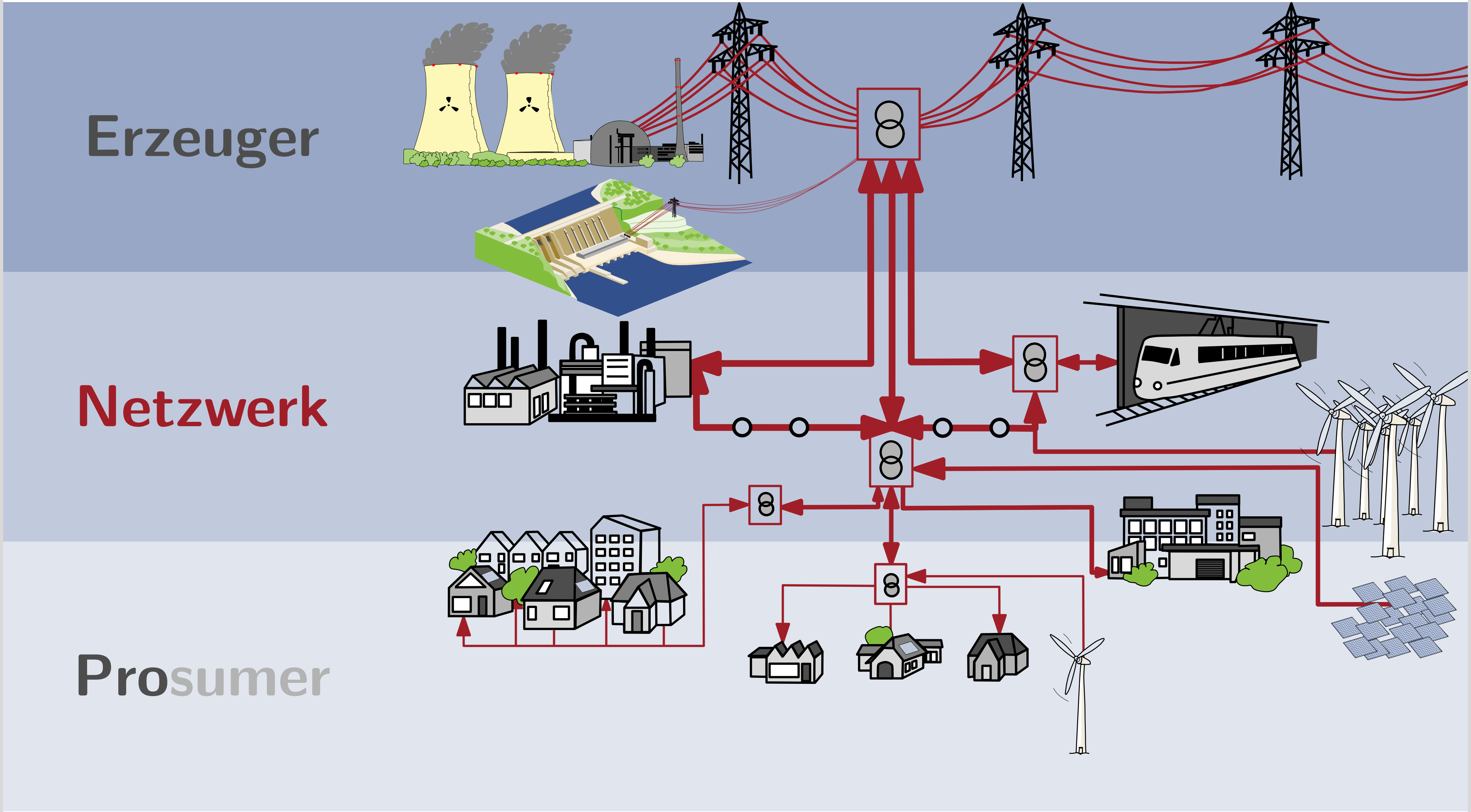
Energieinformatik · Teil 11 (VL2) · 1. Juli, 2020

Franziska Wegner

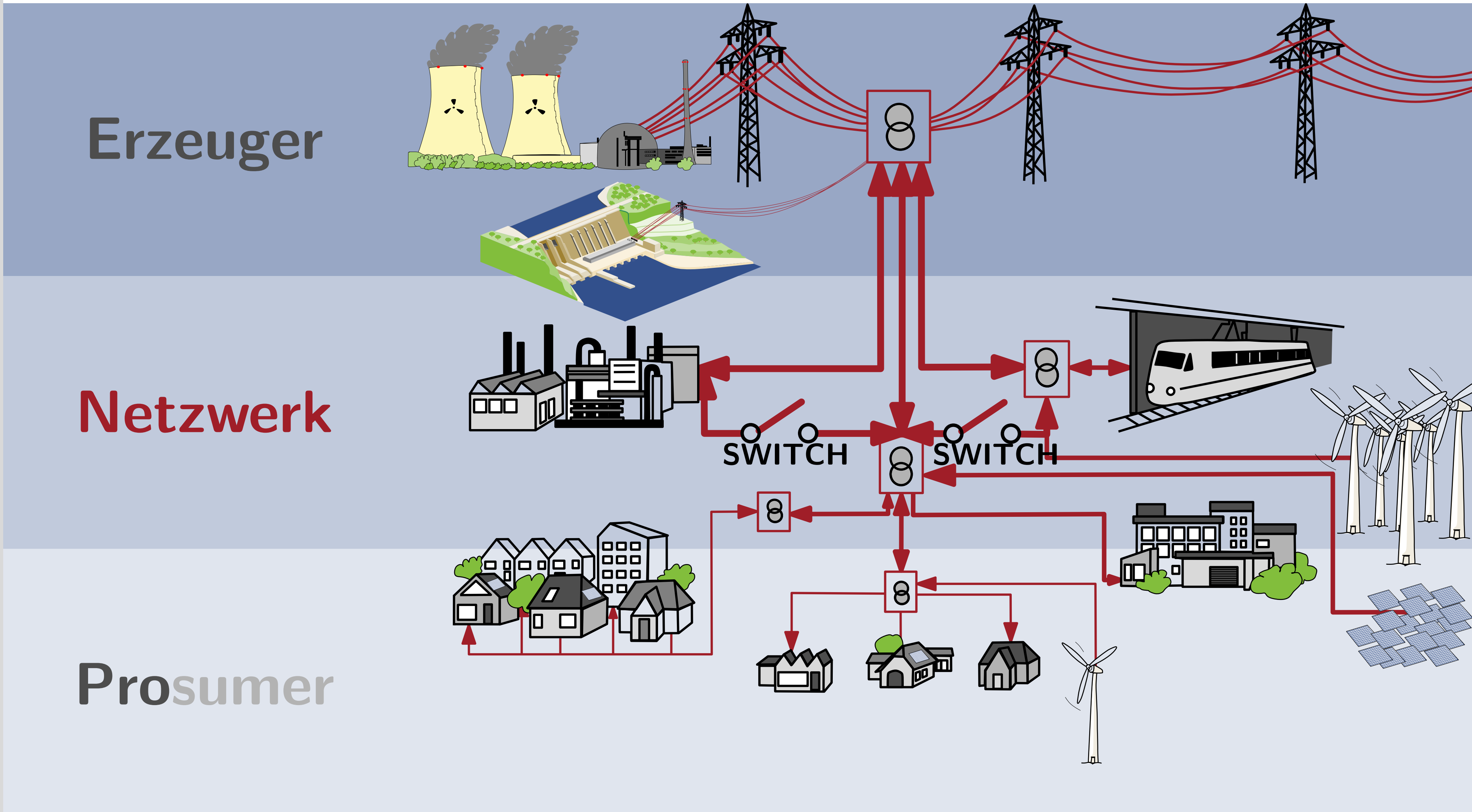
Institut für Theoretische Informatik · Lehrstuhl Algorithmik



Aktuelle Entwicklungen im Energienetz



Aktuelle Entwicklungen im Energienetz



Eingabe

- Sei $G = (V, E)$ ein beliebig gerichteter Graph und $\overleftrightarrow{G} = (V, \overleftrightarrow{E})$ der zugrundeliegende ungerichtete Graph
- Menge von Knoten V (auch Busse genannt) mit Erzeugern $V_G \subseteq V$, Verbrauchern $V_D \subseteq V \setminus V_G$, und Zwischenknoten $V \setminus (V_G \cup V_D)$
- Wir bezeichnen $\overleftrightarrow{E}$ als die darunterliegenden ungerichtete Kantenmenge mit $\overleftrightarrow{e} \in \overleftrightarrow{E}$, sodass $\overleftrightarrow{(u, v)} = \overleftrightarrow{(v, u)}$
- Netzwerk $\mathcal{N} = (G, V_G, V_D, \mathrm{cap}, b, \underline{p_d})$
  thermische Leitungsbeschränkungen $\mathrm{cap}\colon E \to \mathbb{R}_{\geq 0}$,
  Suszeptanzen $b\colon E \to \mathbb{R}_{\geq 0}$,
  untere Schranken der Verbraucher $\underline{p_d}\colon V_D \to \mathbb{R}_{\geq 0}$

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Das Maximale Schaltungsflussproblem für Übertragungsnetze

Gegeben V Bussmenge, $V_D \subseteq V$ Verbrauchermenge (mit Kapazitäten),
$V_G \subseteq V$ Erzeugermenge (mit Kapazitäten)
E Leitungsmenge (mit **Impedanz, Suszeptanz, Kapazität**)

Eingaben

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Das Maximale Schaltungsflussproblem für Übertragungsnetze

Gegeben V Bussmenge, $V_D \subseteq V$ Verbrauchermenge (mit Kapazitäten),
$V_G \subseteq V$ Erzeugermenge (mit Kapazitäten)
E Leitungsmenge (mit **Impedanz, Suszeptanz, Kapazität**)

gesucht für jede Kante: **ob die Leitung geschaltet wurde**

Variablen

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Das Maximale Schaltungsflussproblem für Übertragungsnetze

Gegeben V Bussmenge, $V_D \subseteq V$ Verbrauchermenge (mit Kapazitäten),
$V_G \subseteq V$ Erzeugermenge (mit Kapazitäten)
E Leitungsmenge (mit **Impedanz, Suszeptanz, Kapazität**)

gesucht für jede Kante: **ob die Leitung geschaltet wurde**

maximiere **Leistungserzeugung**

Zielfunktion

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

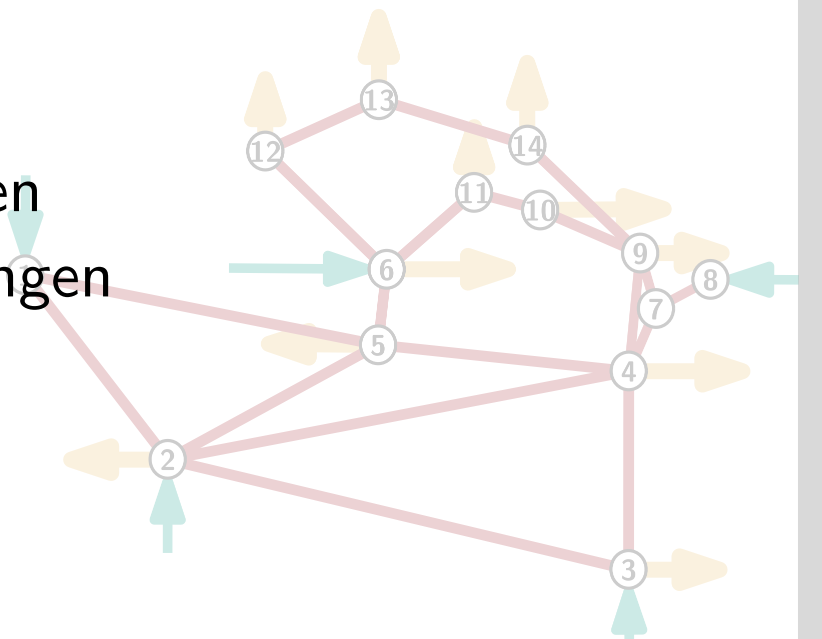
Das Maximale Schaltungsflussproblem für Übertragungsnetze

Gegeben V Bussmenge, $V_D \subseteq V$ Verbrauchermenge (mit Kapazitäten),
$V_G \subseteq V$ Erzeugermenge (mit Kapazitäten)
E Leitungsmenge (mit **Impedanz, Suszeptanz, Kapazität**)

gesucht für jede Kante: **ob die Leitung geschaltet wurde**

maximiere **Leistungserzeugung**

unter Leitungskapazitätseinschränkungen
Verbrauchskapazitätseinschränkungen
Leistungsflusseinschränkungen



Das Maximale Schaltungsflussproblem für Übertragungsnetze

Karlsruher Institut für Technologie

Gegeben V Bussmenge, $V_D \subseteq V$ Verbrauchermenge (mit Kapazitäten),
$V_G \subseteq V$ Erzeugermenge (mit Kapazitäten)
E Leitungsmenge (mit **Impedanz, Suszeptanz, Kapazität**)

Die **AC** Flusserhaltung ist ein **Teilproblem** des MTSF-Problems.

AC Flusserhaltung ist bereits **NP schwer** auf **Bäumen**.
[Lehmann et al., 2015]

unter Leitungskapazitätseinschränkungen
Verbrauchskapazitätseinschränkungen
Leistungsflusseinschränkungen

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Das Maximale Schaltungsflussproblem für Übertragungsnetze

Gegeben V Bussmenge, $V_D \subseteq V$ Verbrauchermenge (mit Kapazitäten),
$V_G \subseteq V$ Erzeugermenge (mit Kapazitäten)
E Leitungsmenge (mit **Impedanz, Suszeptanz, Kapazität**)

Die **AC** Flusserhaltung ist ein **Teilproblem** des MTSF-Problems.

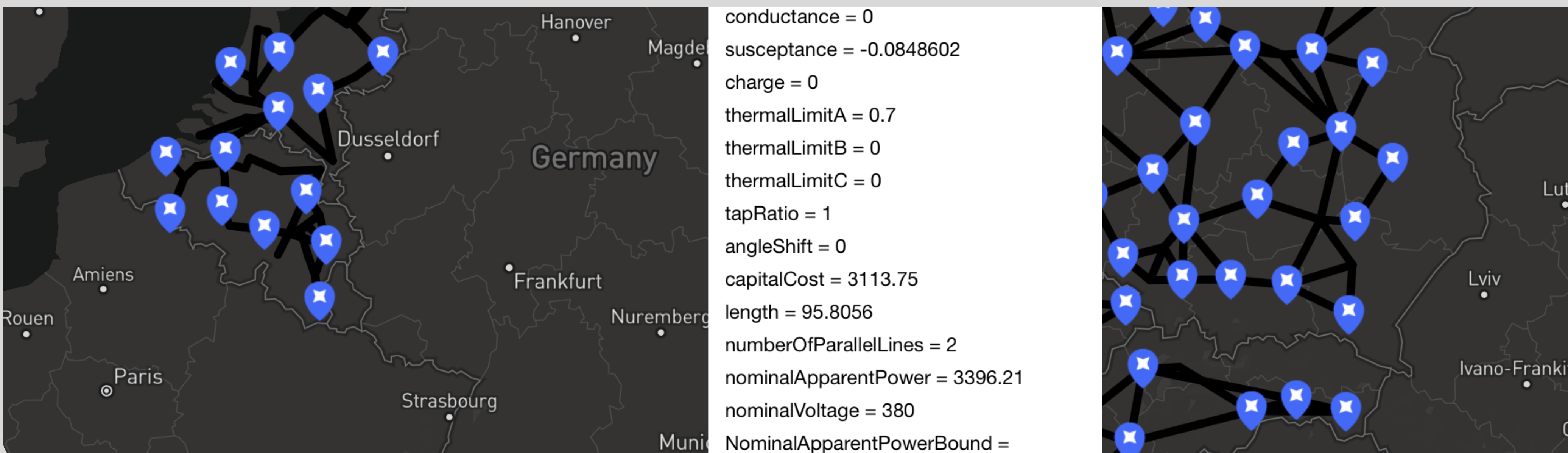
AC Flusserhaltung ist bereits **NP schwer** auf **Bäumen**.

$\rightarrow$ Energienetze sind nicht einfach.
$\rightarrow$ **Linearisierte AC** Flusserhaltung ist **einfach** zu lösen.

Verbrauchskapazitätseinschränkungen
Leistungsflusseinschränkungen

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

(Elektrische) Flüsse und deren mathematische Formulierung



Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Gültige Flüsse

- Ein Fluss ist eine Funktion $f\colon E \cup E^{-1} \to \mathbb{R}$ mit Schiefsymmetrie $f(u,v) = -f(v,u)$ für alle $(u,v) \in E$
- Der Nettofluss $f_{\text{net}}(u) \coloneqq \sum_{\{u,v\} \in \overleftrightarrow{E}} f(u,v)$
- Fluss f erfüllt die folgenden Flusserhaltungseigenschaften, die ähnlich zur Kirchhoff'schen Knotenregel (KCL) sind

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

- Fluss f wird als *gültig* bezeichnet, wenn

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

- Flusswert $F(\mathcal{N}, f)$ vom Fluss f auf $\mathcal{N}$ ist definiert durch

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

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Das Maximale Fluss Problem (MFP)

- Flusswert $F(\mathcal{N}, f)$ vom Fluss f auf $\mathcal{N}$ ist definiert durch
$$\sum_{u \in V_G} f_{\text{net}}(u)$$
- Der maximale Fluss (MF) besitzt den Wert
$$\text{OPT}_{\text{MFP}}(\mathcal{N}) = \max F(\mathcal{N}, f),$$
wobei f ein gültiger Fluss ist, wenn

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

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Physikalisch zulässige Flüsse

- Ein zulässiger Fluss vernachlässigt physikalische Gesetzmäßigkeiten
- Die Kirchhoff'sche Maschenregel (KVL) ist eines davon. Diese wird bspw. mittels Potentialen an den Knoten $\theta^v\colon V \to \mathbb{R}$ formuliert
$$b(u,v) \cdot (\theta^v(v) - \theta^v(u) - \theta^v_{\text{shift}}(u,v)) = f(u,v) \quad \forall (u,v) \in E$$
$$\underline{\theta^v}(u) \leq \theta^v(u) \leq \overline{\theta^v}(u) \quad \forall u \in V$$
- In unseren Fällen gibt es keine Transformatoren und es gilt damit $\theta^v_{\text{shift}}(u,v) = 0$

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Das Maximum Power Flow Problem (MPFP)

- Der maximale Power Flow (MPF) besitzt den Wert
$$\mathrm{OPT}_{\mathrm{MPFP}}(\mathcal{N}) = \max F(\mathcal{N}, f),$$
wobei f ein *physikalisch* zulässiger Fluss ist mit

$$
\begin{aligned}
f_{\mathrm{net}}(u) &= 0 && \forall u \in V \setminus (V_G \cup V_D)\\
-\infty \leq f_{\mathrm{net}}(u) &\leq -\underline{p_d}(u) && \forall u \in V_D\\
0 \leq f_{\mathrm{net}}(u) &\leq \infty && \forall u \in V_G\\
|f(u,v)| &\leq \mathrm{cap}(u,v) && \forall (u,v) \in E\\
b(u,v) \cdot \left(\theta^v(v) - \theta^v(u) - \theta^v_{\mathrm{shift}}(u,v)\right) &= f(u,v) && \forall (u,v) \in E\\
\underline{\theta^v}(u) \leq \theta^v(u) &\leq \overline{\theta^v}(u) && \forall u \in V
\end{aligned}
$$

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Das Maximum Power Flow Problem (MPFP)

- Der maximale Power Flow (MPF) besitzt den Wert
  $$\mathrm{OPT}_{\mathrm{MPFP}}(\mathcal{N}) = \max F(\mathcal{N}, f),$$
  wobei f ein *physikalisch* zulässiger Fluss ist mit

$$
\begin{aligned}
f_{\mathrm{net}}(u) &= 0 && \forall u \in V \setminus (V_G \cup V_D)\\
-\infty \leq f_{\mathrm{net}}(u) &\leq -\underline{p_d}(u) && \forall u \in V_D\\
0 \leq f_{\mathrm{net}}(u) &\leq \infty && \forall u \in V_G\\
|f(u,v)| &\leq \mathrm{cap}(u,v) && \forall (u,v) \in E\\
b(u,v) \cdot \left(\theta^{v}(v) - \theta^{v}(u)\right) &= f(u,v) && \forall (u,v) \in E\\
\underline{\theta^{v}}(u) \leq \theta^{v}(u) &\leq \overline{\theta^{v}}(u) && \forall u \in V
\end{aligned}
$$

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

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

- Der Maximum Transmission Switching Flow (MTSF) besitzt den Wert
$$\mathrm{OPT}_{\mathrm{MTSF}}(\mathcal{N}) \coloneqq \max_{S \subseteq \overleftrightarrow{E}} \mathrm{OPT}_{\mathrm{MPFP}}(\mathcal{N} - S),$$
wobei f ein physikalisch zulässiger Fluss ist

$$\begin{aligned}
f_{\text{net}}(u) &= 0 && \forall u \in V \setminus (V_G \cup V_D) \\
-\infty \leq f_{\text{net}}(u) &\leq -\underline{p_d}(u) && \forall u \in V_D \\
0 \leq f_{\text{net}}(u) &\leq \infty && \forall u \in V_G \\
|f(u,v)| &\leq z(u,v) \cdot \mathrm{cap}(u,v) && \forall (u,v) \in E \\
b(u,v) \cdot z(u,v) \cdot \big(\theta^v(v) - \theta^v(u)\big) &= f(u,v) && \forall (u,v) \in E \\
\underline{\theta^v}(u) \leq \theta^v(u) &\leq \overline{\theta^v}(u) && \forall u \in V
\end{aligned}$$

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

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

- Der Maximum Transmission Switching Flow (MTSF) besitzt den Wert
$$\mathrm{OPT}_{\mathrm{MTSF}}(\mathcal{N}) := \max_{S \subseteq \overleftrightarrow{E}} \mathrm{OPT}_{\mathrm{MPFP}}(\mathcal{N} - S),$$
wobei f ein physikalisch zulässiger Fluss ist

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

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

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

Optimierungsproblem MTSF

Instanz: Ein Energienetz $\mathcal{N}$.

Zielfunktion: Finde eine Menge $S \subseteq \overleftrightarrow{E}$ von geswitchten Kanten, sodass $\text{OPT}_{\text{MPF}}(\mathcal{N} - S)$ maximal unter allen möglichen geswitchten Kanten S ist.

Entscheidungsproblem k-MTSF

Instanz: Ein Übertragungsnetz $\mathcal{N}$ und $k \in \mathbb{Q}_{\geq 0}$.

Zielfunktion: Ist es möglich eine Menge von Kanten $S \subseteq \overleftrightarrow{E}$ zu entfernen, sodass es einen physikalischen zuässigen Fluss f in $\mathcal{N} - S$ mit Flusswert $F(\mathcal{N} - S, f) \geq k$ gibt?

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Das Optimal Power Flow (OPF) Problem

[Zimmerman et al., 2011]

Karlsruher Institut für Technologie

- Network $\mathcal{N} = (G, V_G, V_D, \text{cap}, b, \underline{p_d})$

- Das Optimal Power Flow (OPF) besitzt den Wert
  $$\text{OPT}_{\text{OPF}}(\mathcal{N}) := \min \sum_{u \in V_G} \gamma_u\big(f_{\text{net}}(u)\big)$$
  wobei f der physical zulässige Fluss ist
  $$\begin{aligned} f_{\text{net}}(u) &= 0 & \forall u \in V \setminus (V_G \cup V_D) \\ -\infty \leq f_{\text{net}}(u) &\leq -\underline{p_d}(u) & \forall u \in V_D \\ 0 \leq f_{\text{net}}(u) &\leq \infty & \forall u \in V_G \\ |f(u,v)| &\leq \text{cap}(u,v) & \forall (u,v) \in E \\ b(u,v) \cdot \big(\theta^v(v) - \theta^v(u)\big) &= f(u,v) & \forall (u,v) \in E \\ \underline{\theta^v}(u) \leq \theta^v(u) &\leq \overline{\theta^v}(u) & \forall u \in V \end{aligned}$$

Das Optimal Power Flow (OPF) Problem

[Zimmerman et al., 2011]

- Network $\mathcal{N}_{\text{bounded}} = (G, V_G, V_D, \text{cap}, b, \overbrace{\underline{p_g}, \overline{p_g}, \underline{p_d}, \overline{p_d}}^{\in \mathbb{R}})$ wird als bounded bezeichnet
- Erzeugungskostenfunktion $\gamma_u \colon \mathbb{R} \to \mathbb{R}_{\geq 0}$ für alle $u \in V_G$
- Das Optimal Power Flow (OPF) besitzt den Wert
$$\text{OPT}_{\text{OPF}}(\mathcal{N}) := \min \sum_{u \in V_G} \gamma_u\big(f_{\text{net}}(u)\big)$$
wobei f der physical zulässige Fluss ist

$$
\begin{aligned}
f_{\text{net}}(u) &= 0 && \forall u \in V \setminus (V_G \cup V_D) \\
-\overline{p_d}(u) \leq f_{\text{net}}(u) &\leq -\underline{p_d}(u) && \forall u \in V_D \\
\underline{p_g} \leq f_{\text{net}}(u) &\leq \overline{p_g} && \forall u \in V_G \\
\big|f(u,v)\big| &\leq \text{cap}(u,v) && \forall (u,v) \in E \\
b(u,v) \cdot \big(\theta^V(v) - \theta^V(u)\big) &= f(u,v) && \forall (u,v) \in E \\
\underline{\theta^V}(u) \leq \theta^V(u) &\leq \overline{\theta^V}(u) && \forall u \in V
\end{aligned}
$$

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Das Optimal Power Flow (OPF) Problem

[Zimmerman et al., 2011]

- Network $\mathcal{N}_{\text{bounded}} = (G, V_G, V_D, \text{cap}, b, \overbrace{\underline{p_g}, \overline{p_g}, \underline{p_d}, \overline{p_d}}^{\in \mathbb{R}})$ wird als bounded bezeichnet
- Erzeugungskostenfunktion $\gamma_u \colon \mathbb{R} \to \mathbb{R}_{\geq 0}$ für alle $u \in V_G$
- Das Optimal Power Flow (OPF) besitzt den Wert

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

wobei f der physical zulässige Fluss ist

$$\begin{aligned}
f_{\text{net}}(u) &= 0 & \forall u \in V \setminus (V_G \cup V_D) \\
-p_d(u) = -\overline{p_d}(u) \leq f_{\text{net}}(u) &\leq -\underline{p_d}(u) = -p_d(u) & \forall u \in V_D \\
\underline{p_g} \leq f_{\text{net}}(u) &\leq \overline{p_g} & \forall u \in V_G \\
|f(u,v)| &\leq \text{cap}(u,v) & \forall (u,v) \in E \\
b(u,v) \cdot \big(\theta^v(v) - \theta^v(u) \big) &= f(u,v) & \forall (u,v) \in E \\
\underline{\theta^v}(u) \leq \theta^v(u) &\leq \overline{\theta^v}(u) & \forall u \in V
\end{aligned}$$

Das Optimal Power Flow (OPF) Problem

[Zimmerman et al., 2011]

- Network $\mathcal{N}_{\text{bounded}} = (G, V_G, V_D, \text{cap}, b, \overbrace{\underline{p_g}, \overline{p_g}, \underline{p_d}, \overline{p_d}}^{\in \mathbb{R}})$ wird als bounded bezeichnet
- Erzeugungskostenfunktion $\gamma_u : \mathbb{R} \to \mathbb{R}_{\geq 0}$ für alle $u \in V_G$
- Das Optimal Power Flow (OPF) besitzt den Wert

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

wobei f der physical zulässige Fluss ist

$$\begin{aligned}
f_{\text{net}}(u) &= 0 && \forall u \in V \setminus (V_G \cup V_D) \\
f_{\text{net}}(u) &= -p_d(u) && \forall u \in V_D \\
\underline{p_g}(u) \leq f_{\text{net}}(u) &\leq \overline{p_g}(u) && \forall u \in V_G \\
\big|f(u,v)\big| &\leq \text{cap}(u,v) && \forall (u,v) \in E \\
b(u,v) \cdot \big(\theta^v(v) - \theta^v(u)\big) &= f(u,v) && \forall (u,v) \in E \\
\underline{\theta^v}(u) \leq \theta^v(u) &\leq \overline{\theta^v}(u) && \forall u \in V
\end{aligned}$$

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

Karlsruher Institut für Technologie

- Das Optimal Transmission Switching (OTS) besitzt den Wert

$$\mathrm{OPT}_{\mathrm{OTS}}(\mathcal{N}) \coloneqq \min_{S \subseteq \overleftrightarrow{E}} \mathrm{OPF}(\mathcal{N} - S)$$

wobei f ein physikalisch zulässiger Fluss ist mit

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

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

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

Karlsruher Institut für Technologie

Optimierungsproblem OTS

Instanz: Ein Energienetz $\mathcal{N}_{\text{bounded}}$.

Zielfunktion: Finde eine Menge $S \subseteq \overleftrightarrow{E}$ von geschalteten Kanten, sodass $\text{OPT}_{\text{OPF}}(\mathcal{N} - S)$ minimum über alle möglichen geschalteten Kanten S ist.

Entscheidungsproblem kOTS

Instanz: Ein Energienetz $\mathcal{N}_{\text{bounded}}$ und $k \in \mathbb{Q}_{\geq 0}$.

Zielfunktion: Ist es möglich eine Kantenmenge $S \subseteq \overleftrightarrow{E}$ zu entfernen, sodass ein physikalischer zulässiger Fluss f in $\mathcal{N} - S$ mit Kosten $\sum_{u \in V_G} \gamma_u\left(f_{\text{net}}(u)\right) \leq k$ erreicht wird?

Verbindung zum DC-Modell

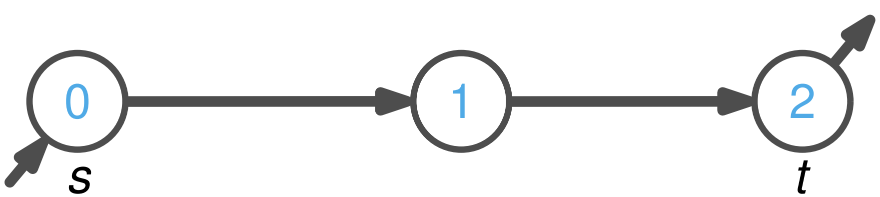
Karlsruher Institut für Technologie

- Der Leistungsfluss verhält sich wie der Strom in DC-Netzwerken
- Die Gesamtsuszeptanz $B_{\text{tot}} = -\frac{1}{X}$ ($-\frac{1}{X}$, da $G = R = 0$ in der DC-Approximation, sonst $-\frac{X}{R^2+X^2}$) verhalten sich wie der elektrische Leitwert $G = \frac{1}{R}$
- Die Spannungswinkeldifferenzen $\Delta\theta^{v}$ verhalten sich wie Spannungen U

$$I_{\text{tot}} = \frac{U_{\text{tot}}}{R_{\text{tot}}} = U_{\text{tot}} \cdot G_{\text{tot}} = \Delta\theta^{v}_{\text{tot}} \cdot B_{\text{tot}} = P_{\text{tot}}$$

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

Serienschaltkreis

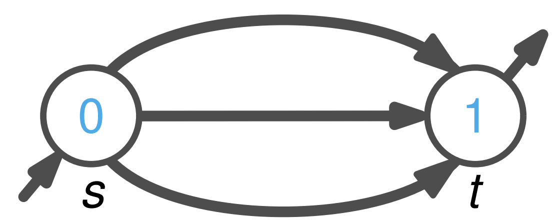


$$B_{\text{tot}} = b(s,t) = \frac{1}{\sum_{k=1}^{n} \frac{1}{b_k}} = \frac{1}{2}$$

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

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

Parallelschaltkreis

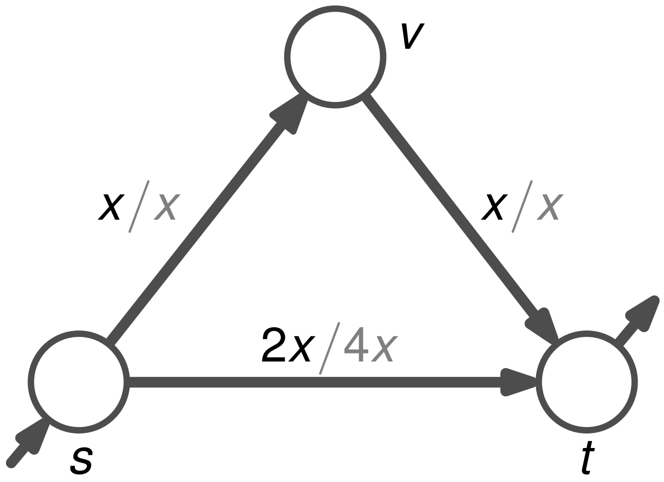


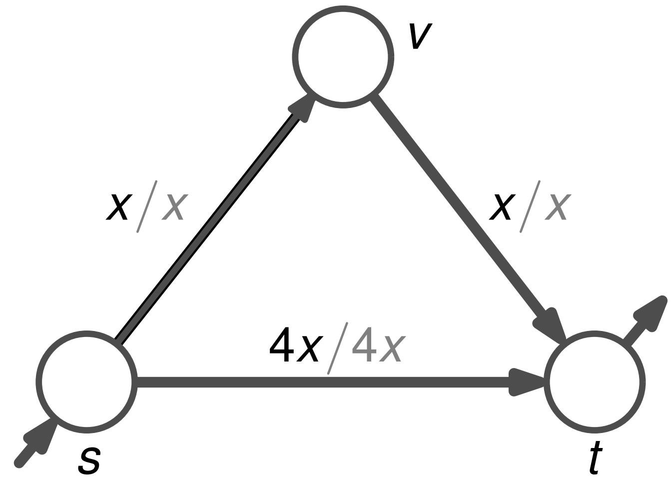
$$B_{\text{tot}} = b(s,t) = \sum_{k=1}^{n} b_k = 3$$

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

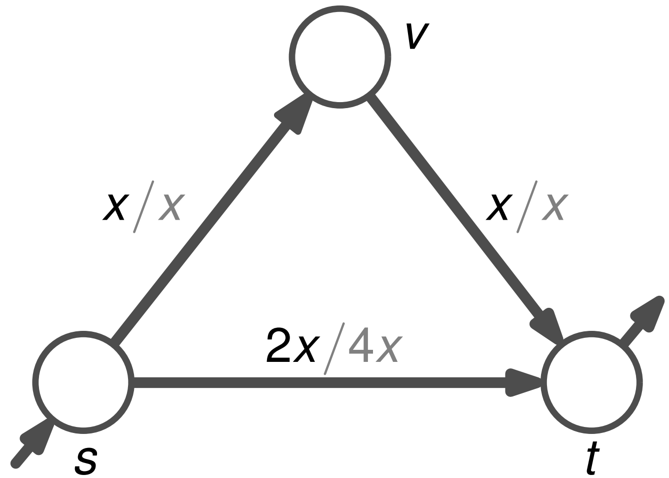
$$\Delta\theta^{v}_{\text{tot}} = \Delta\theta^{v}(s,t) = \min_{(s,t)^i}(\Delta\theta^{v}_i(s,t)) = 1$$

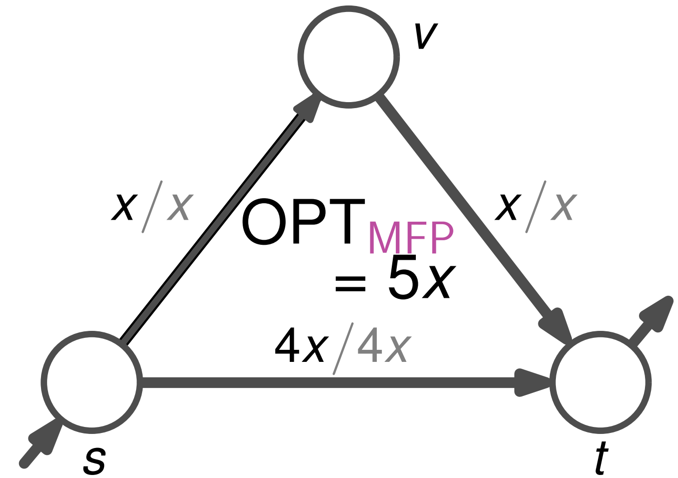
Das MTSF Problem





Das MTSF Problem



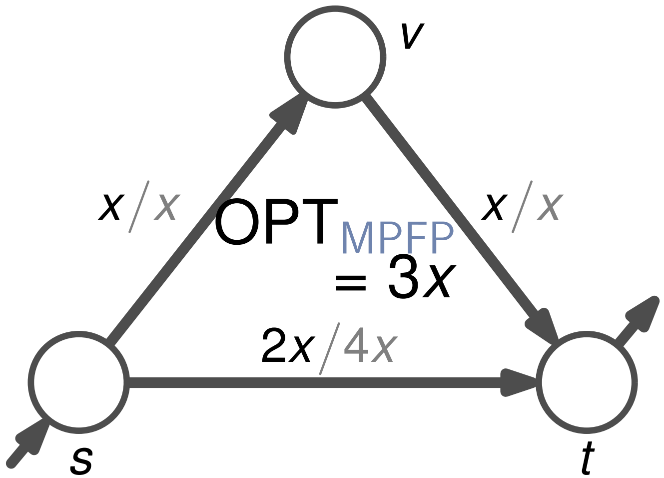


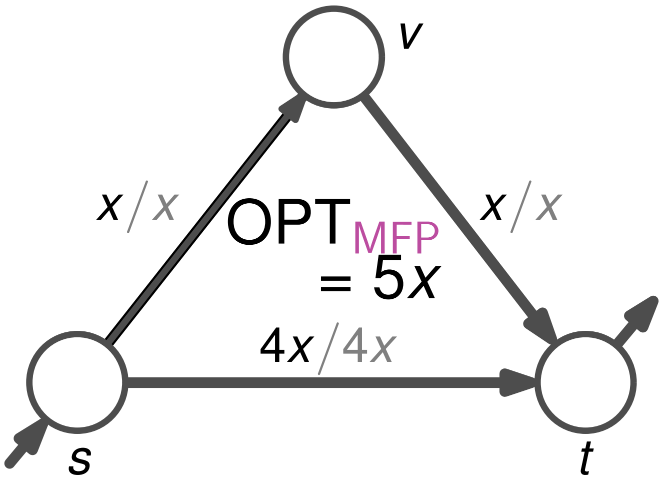
Flussmodell

obere Schranke

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Das MTSF Problem





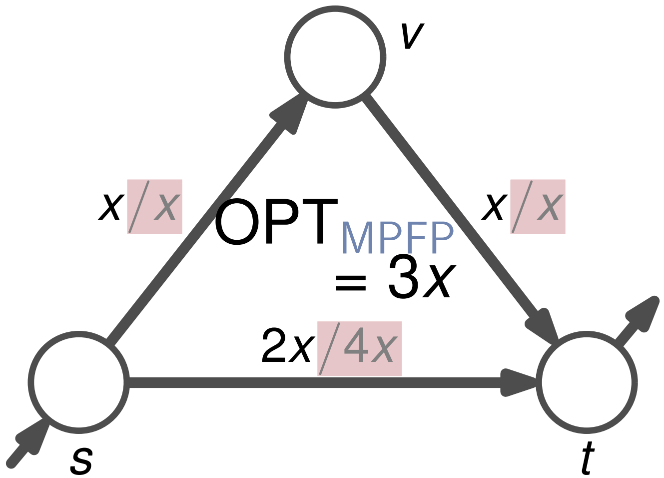
physikalisches Modell
(AC-Linearisierung)

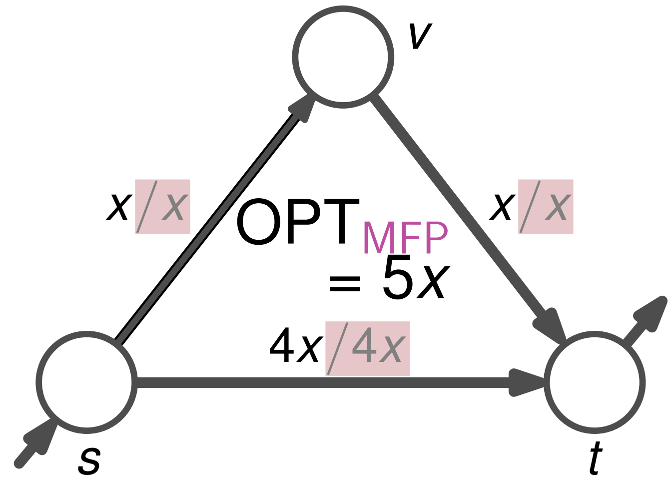
untere Schranke

Flussmodell

obere Schranke

Das MTSF Problem





physikalisches Modell
(AC-Linearisierung)

Flussmodell

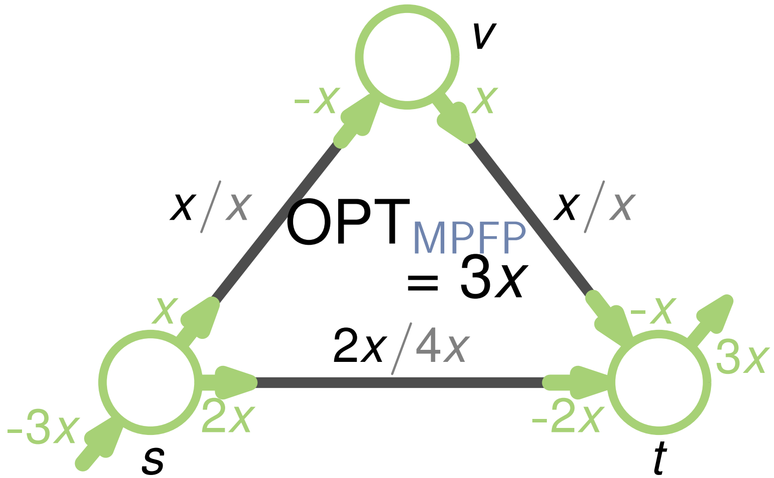
untere Schranke

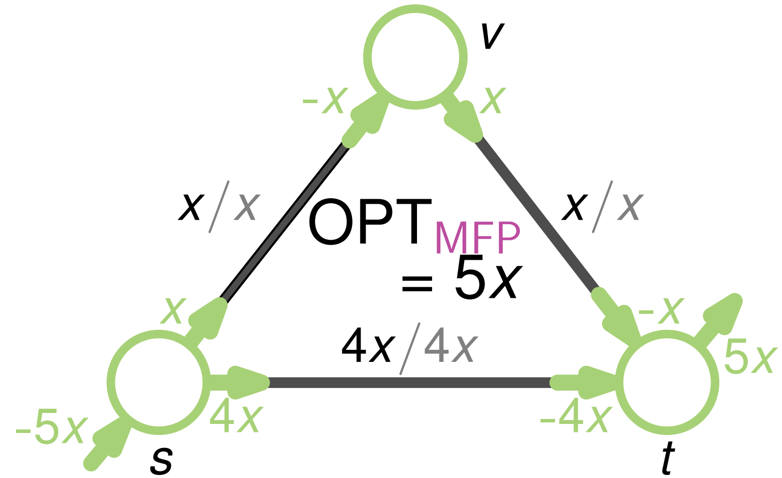
obere Schranke

Kapazitätseinschränkungen

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Das MTSF Problem





physikalisches Modell
(AC-Linearisierung)

Flussmodell

untere Schranke

obere Schranke

Kapazitätseinschränkungen

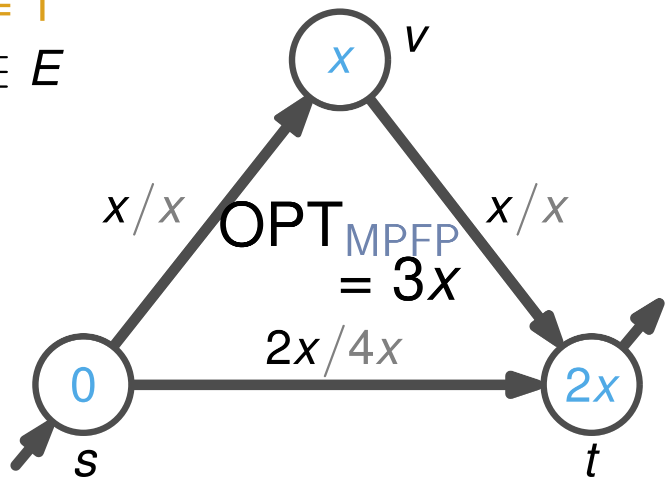
Kirchhoff'sche Knotenregel (KCL)

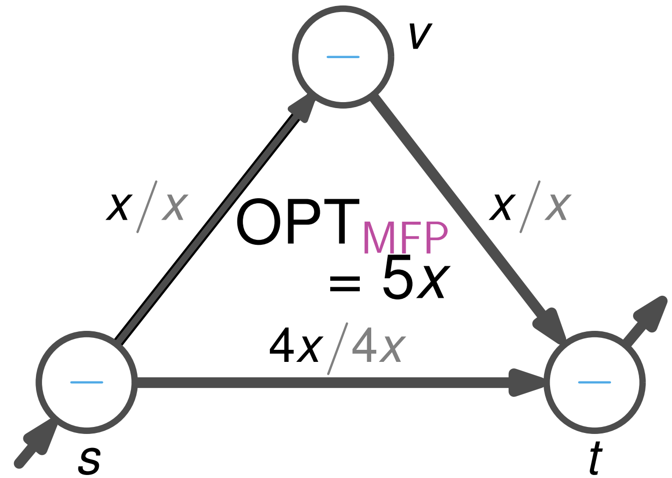
Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Das MTSF Problem

Karlsruher Institut für Technologie

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





physikalisches Modell
(AC-Linearisierung)

Flussmodell

untere Schranke

obere Schranke

Kapazitätseinschränkungen

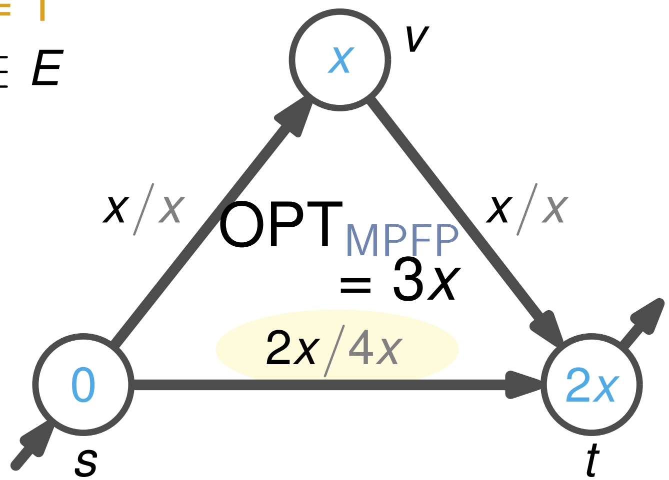
Kirchhoff'sche Knotenregel (KCL)

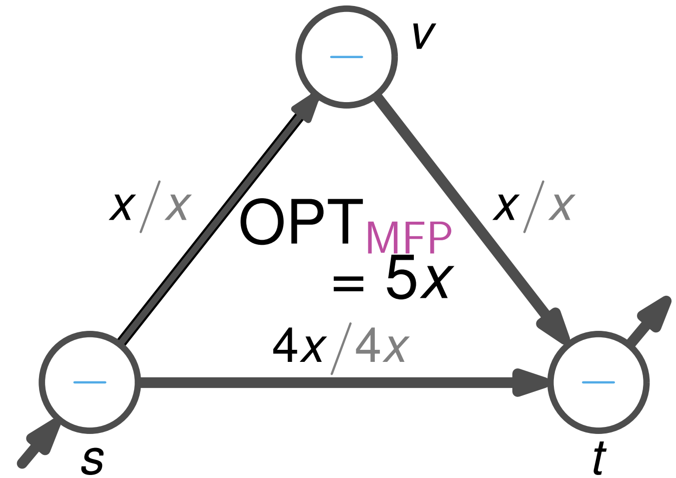
DC Leistungsflusseinschränkung

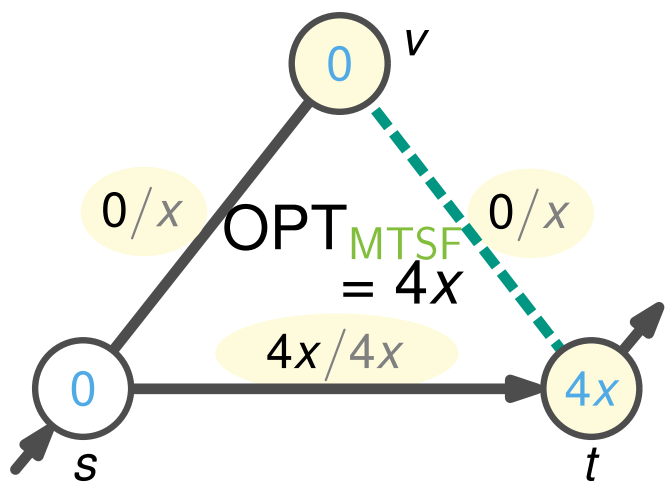
$\forall (u,v) \in E\colon\ f(u,v) = b(u,v)\big(\theta^{v}(v) - \theta^{v}(u)\big)$

Das MTSF Problem

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





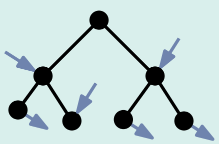


$$\forall (u, v) \in E\colon\ f(u, v) = b(u, v)\left(\theta^{v}(v) - \theta^{v}(u)\right)$$

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Das MTSF Problem

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

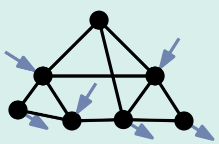


Phy. Modell (MPFP)	=	Maximaler Schaltfluss (MTSFP)	=	Flussmodell (MFP)

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

Das MTSF Problem

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



Phy. Modell (MPFP)	$\leq$	Maximaler Schaltfluss (MTSFP)	$\leq$	Flussmodell (MFP)

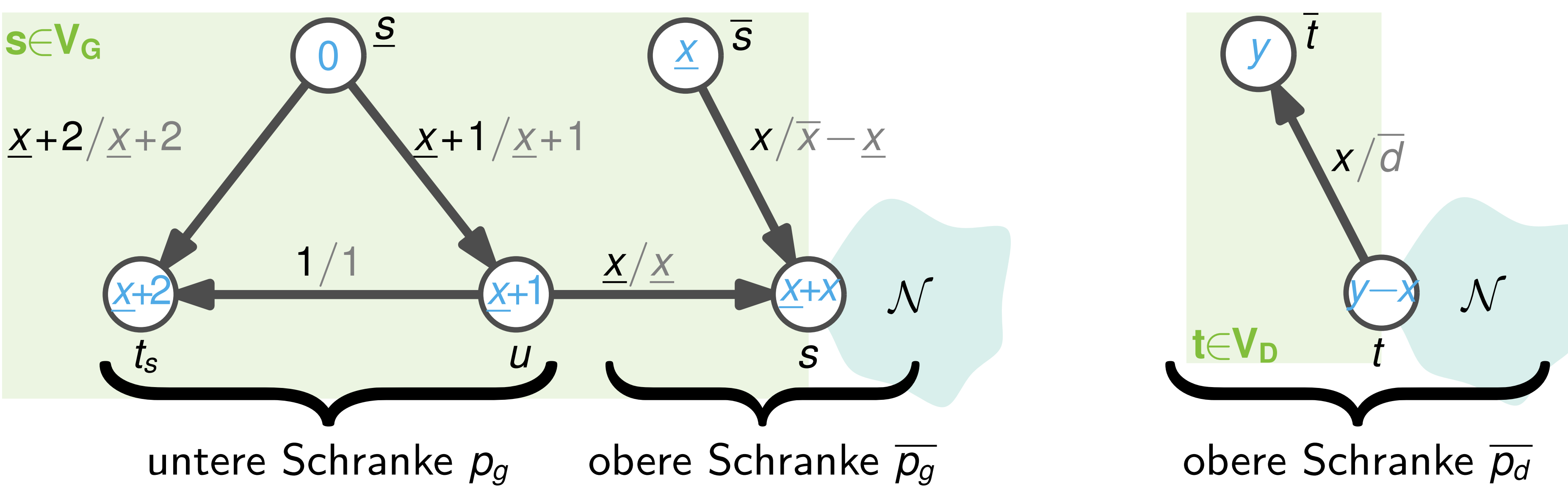
$$\forall (u,v) \in E\colon f(u,v) = b(u,v)\big(\theta^v(v) - \theta^v(u)\big)$$

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Netzwerkmodellierung – Beschränkt zu Unbeschränkt

- Transformation von einem beschränkten Netzwerk $\mathcal{N}_{\text{bounded}} = (G, V_G, V_D, \text{cap}, b, \underline{p_g}, \overline{p_g}, \underline{p_d}, \overline{p_d})$ in ein unbeschränktes Netzwerk $\mathcal{N} = (G, V_G, \overline{V_D}, \text{cap}, b, \underline{p_d})$

Modelliere Erzeugerschranken in $\mathcal{N}$ **Modelliere Verbraucherschranken in $\mathcal{N}$**



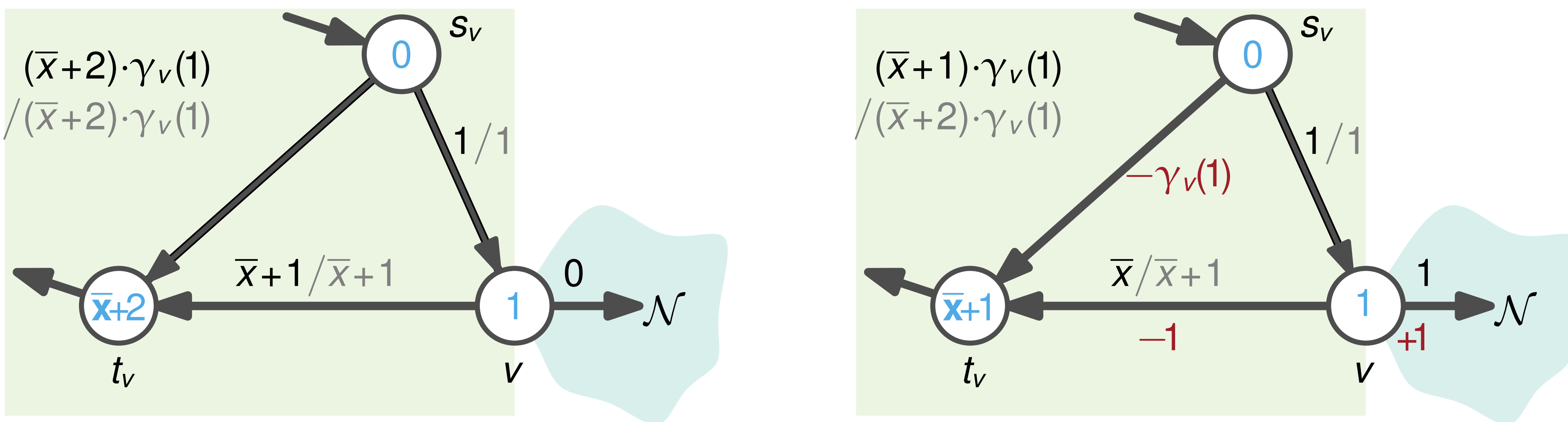
Lemma 1 [S.343; Grastien et al., 2018]

Jedes beschränkte MTSF kann transformiert werden in ein unbeschränktes MTSF auf ein Netzwerk mit einer Größe linear in $|V|$ und $|E|$.

Netzwerkmodellierung – OTS zu MTSF

- OTS-Instanz $\mathcal{N} = (G, V_G, V_D, \text{cap}, b, \underline{p_g}, \overline{p_g}, p_d)$
- $\gamma_v(1)$ Kosten pro erzeugter Leistungseinheit
- $b(s_v, t_v) \coloneqq \gamma_v(1)$, $b(s_v, v) \coloneqq b(v, t_v) \coloneqq 1$

Knoten v speist keine Leistung in das Netzwerk $\mathcal{N}$ ein | **Knoten v speist Leistung in $\mathcal{N}$ ein**



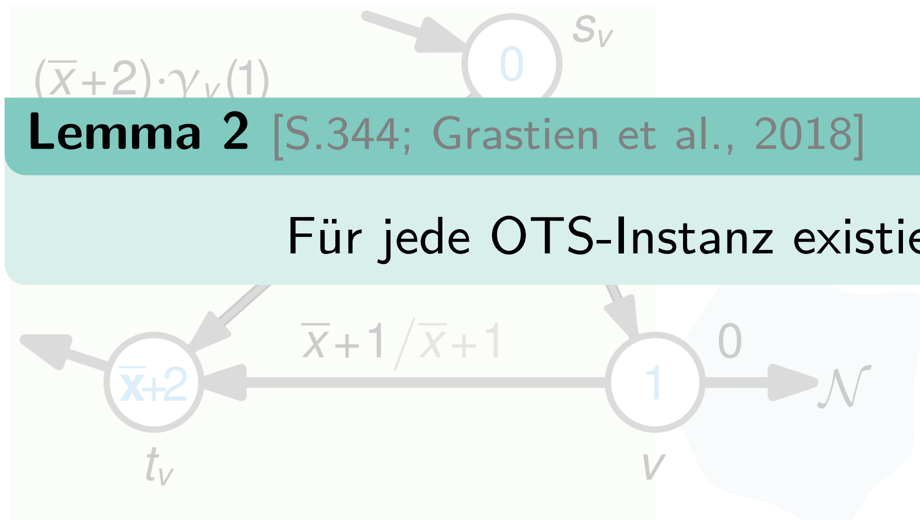
- Zulässiger Fluss in $\mathcal{N}$ mit Kosten k $\rightarrow$ zulässiger Fluss in $\mathcal{N}'$ mit Flusswert $M - k$
- $M = \sum_{v \in V_G} \big((\overline{x} + 2) \cdot \gamma_v(1) + \overline{x}_v + 1\big)$

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

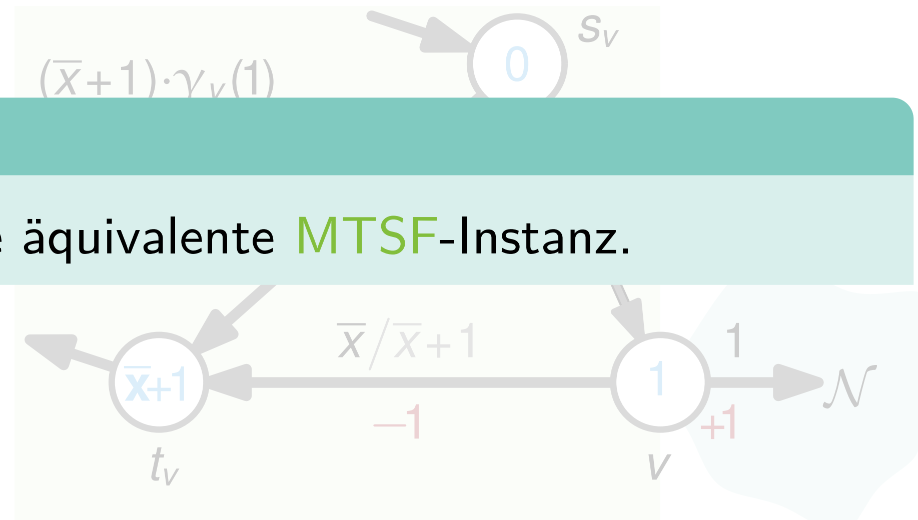
Netzwerkmodellierung – OTS zu MTSF

- OTS-Instanz $\mathcal{N} = (G, V_G, V_D, \text{cap}, b, \underline{p_g}, \overline{p_g}, p_d)$
- $\gamma_v(1)$ Kosten pro erzeugter Leistungseinheit
- $b(s_v, t_v) := \gamma_v(1), b(s_v, v) := b(v, t_v) := 1$

Knoten v speist keine Leistung in das Netzwerk $\mathcal{N}$ ein



Knoten v speist Leistung in $\mathcal{N}$ ein



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

Für jede OTS-Instanz existiert eine äquivalente MTSF-Instanz.

- Zulässiger Fluss in $\mathcal{N}$ mit Kosten k $\rightarrow$ zulässiger Fluss in $\mathcal{N}'$ mit Flusswert $M - k$
- $M = \sum_{v \in V_G} \left((\overline{x} + 2) \cdot \gamma_v(1) + \overline{x}_v + 1 \right)$

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Überblick über MTSF Ergebnisse

Graphenstruktur	Komplexität	Algorithmen

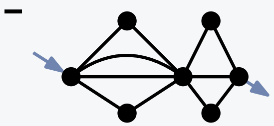
Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Überblick über MTSF Ergebnisse

Karlsruher Institut für Technologie

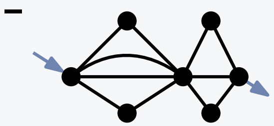
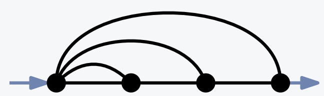
easy

ein Erzeuger, ein Verbraucher

Graphenstruktur	Komplexität	Algorithmen
Penrose-Minoren-freie Graphen 	Polynomialzeit lösbar	DTP

Überblick über MTSF Ergebnisse

Karlsruher Institut für Technologie

Graphenstruktur	Komplexität	Algorithmen
Penrose-Minoren-freie Graphen 	Polynomialzeit lösbar	DTP ✓
Serienparallele Graphen 	NP-schwer	✗

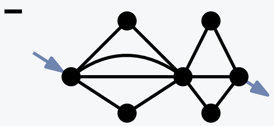
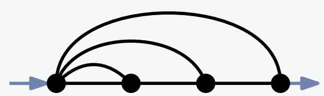
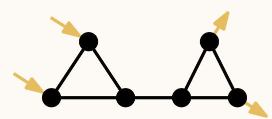
ein Erzeuger, ein Verbraucher

Komplexität

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Überblick über MTSF Ergebnisse

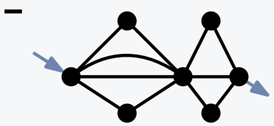
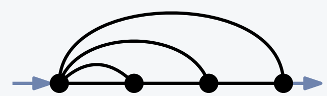
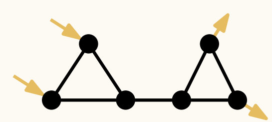
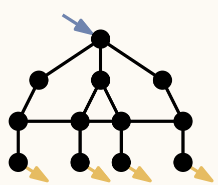
Komplexität

	Graphenstruktur		Komplexität	Algorithmen
ein Erzeuger, ein Verbraucher	Penrose-Minoren-freie Graphen		Polynomialzeit lösbar	DTP ✓
ein Erzeuger, ein Verbraucher	Serienparallele Graphen		NP-schwer	✗
beliebige Erzeuger, beliebige Verbraucher	Kakteen mit Maximalgrad 3		NP-schwer [Lehmann et al., 2014]	2-approx. ✓

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Überblick über MTSF Ergebnisse

Komplexität

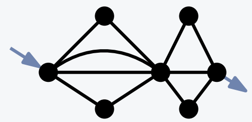
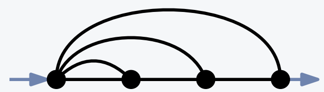
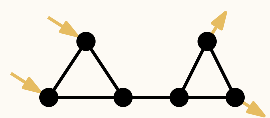
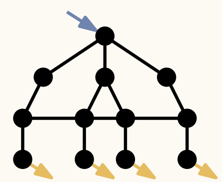
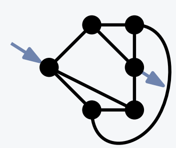
	Graphenstruktur		Komplexität	Algorithmen
ein Erzeuger, ein Verbraucher	Penrose-Minoren-freie Graphen		Polynomialzeit lösbar	DTP ✓
	Serienparallele Graphen		NP-schwer	✗
beliebige Erzeuger, beliebige Verbraucher	Kakteen mit Maximalgrad 3		NP-schwer [Lehmann et al., 2014]	2-approx. ✓
	2-Level-Bäume		NP-schwer [Lehmann et al., 2014]	✗

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Karlsruher Institut für Technologie

Überblick über MTSF Ergebnisse

Komplexität

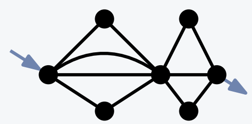
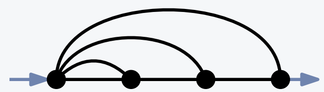
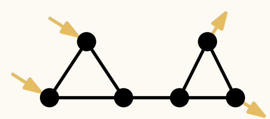
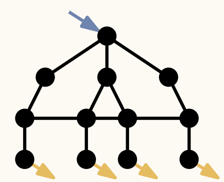
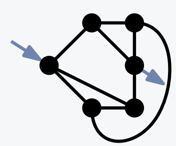
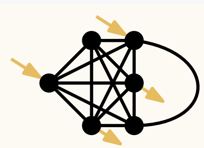
	Graphenstruktur		Komplexität	Algorithmen
ein Erzeuger, ein Verbraucher	Penrose-Minoren-freie Graphen		Polynomialzeit lösbar	DTP ✓
ein Erzeuger, ein Verbraucher	Serienparallele Graphen		NP-schwer	✗
beliebige Erzeuger, beliebige Verbraucher	Kakteen mit Maximalgrad 3		NP-schwer [Lehmann et al., 2014]	2-approx. ✓
beliebige Erzeuger, beliebige Verbraucher	2-Level-Bäume		NP-schwer [Lehmann et al., 2014]	✗
	Planar Graphen mit Maximalgrad 3		stark NP-schwer [Lehmann et al., 2014]	✗

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Überblick über MTSF Ergebnisse

Karlsruher Institut für Technologie

Komplexität

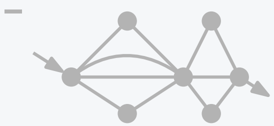
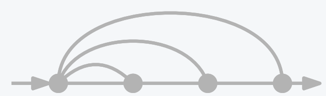
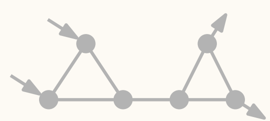
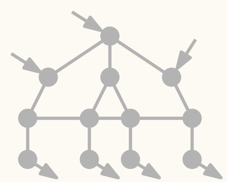
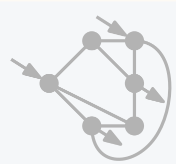
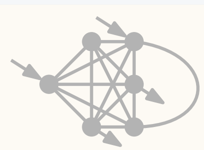
	Graphenstruktur		Komplexität	Algorithmen
ein Erzeuger, ein Verbraucher	Penrose-Minoren-freie Graphen		Polynomialzeit lösbar	DTP ✓
	Serienparallele Graphen		NP-schwer	✗
beliebige Erzeuger, beliebige Verbraucher	Kakteen mit Maximalgrad 3		NP-schwer [Lehmann et al., 2014]	2-approx. ✓
	2-Level-Bäume		NP-schwer [Lehmann et al., 2014]	✗
	Planar Graphen mit Maximalgrad 3		stark NP-schwer [Lehmann et al., 2014]	✗
$\lvert V_G \rvert = 2$, $\lvert V_D \rvert = 2$	Beliebige Graphen		nicht-APX [Lehmann et al., 2014]	✗

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Überblick über MTSF Ergebnisse

Karlsruher Institut für Technologie

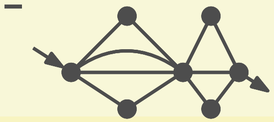
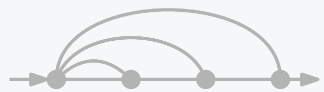
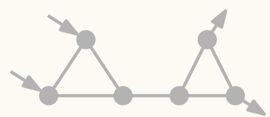
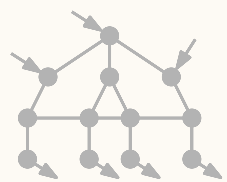
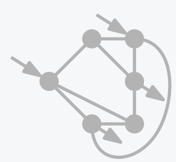
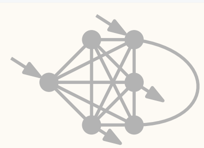
Komplexität ↓

	Graphenstruktur		Komplexität	Algorithmen
ein Erzeuger, ein Verbraucher	Penrose-Minoren-freie Graphen		Polynomialzeit lösbar	DTP ✓
	Serienparallele Graphen		NP-schwer	✗
beliebige Erzeuger, beliebige Verbraucher	Kakteen mit Maximalgrad 3		NP-schwer [Lehmann et al., 2014]	2-approx. ✓
	2-Level-Bäume		NP-schwer [Lehmann et al., 2014]	✗
	Planare Graphen mit Maximalgrad 3		stark NP-schwer [Lehmann et al., 2014]	✗
$\lvert V_G\rvert=2$, $\lvert V_D\rvert=2$	Beliebige Graphen		nicht-APX [Lehmann et al., 2014]	✗

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Überblick über MTSF Ergebnisse

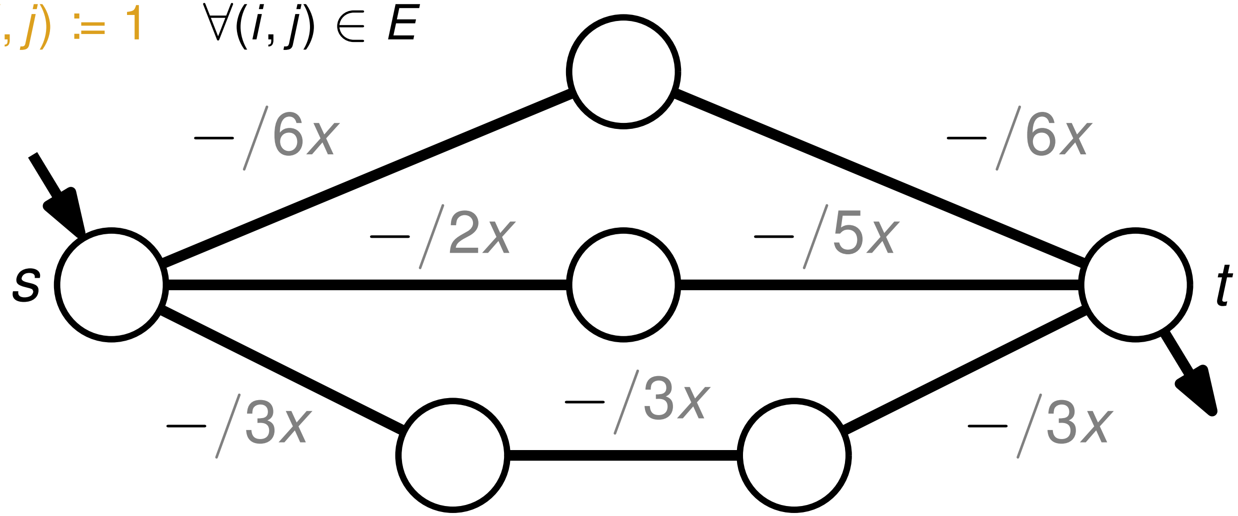
Komplexität

	Graphenstruktur		Komplexität	Algorithmen
ein Erzeuger, ein Verbraucher	Penrose-Minoren-freie Graphen		Polynomialzeit lösbar	DTP
	Serienparallele Graphen		NP-schwer	✗
beliebige Erzeuger, beliebige Verbraucher	Kakteen mit Maximalgrad 3		NP-schwer [Lehmann et al., 2014]	2-approx.
	2-Level-Bäume		NP-schwer [Lehmann et al., 2014]	✗
	Planare Graphen mit Maximalgrad 3		stark NP-schwer [Lehmann et al., 2014]	✗
$\lvert V_G \rvert = 2$, $\lvert V_D \rvert = 2$	Beliebige Graphen		nicht-APX [Lehmann et al., 2014]	✗

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

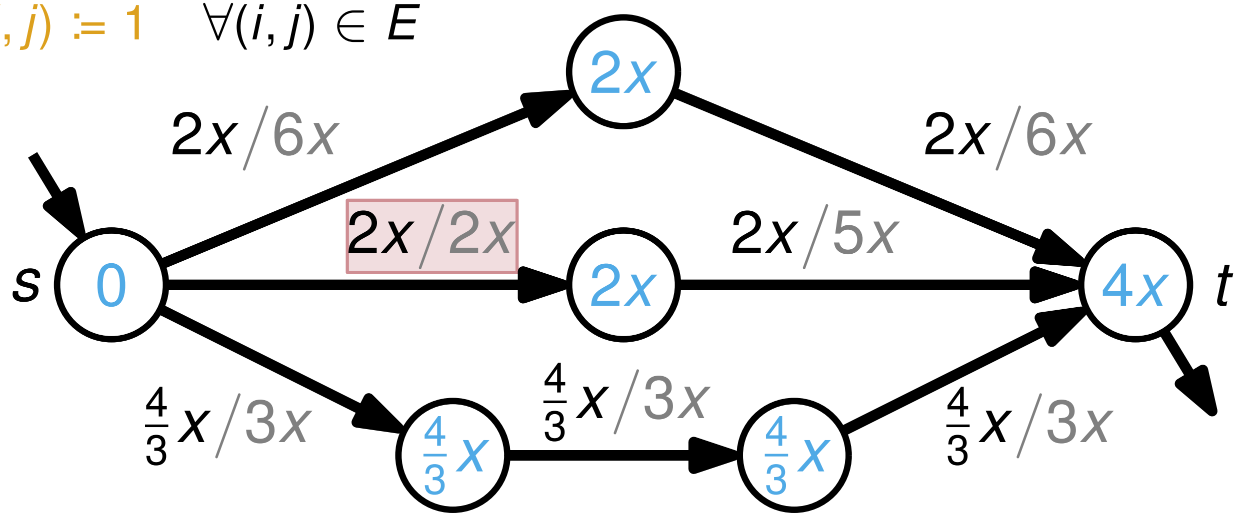
Schalten auf parallelen Pfaden

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



Schalten auf parallelen Pfaden

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



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

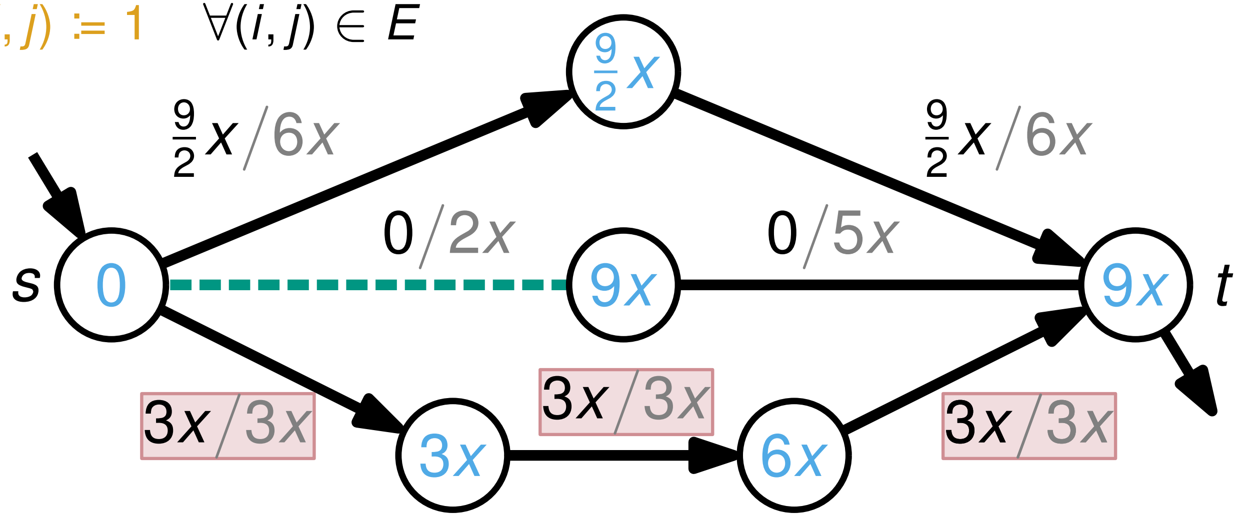
$\Delta\theta^v(\pi) \coloneqq \underbrace{\|\pi\|_b}_{\text{elektrische Distanz}} \cdot \min_{e\in\pi} \mathrm{cap}(e)$

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

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Schalten auf parallelen Pfaden

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



$$\Delta\theta^v(\pi) \coloneqq \underbrace{\|\pi\|_b}_{\text{elektrische Distanz}} \cdot \min_{e\in\pi} \operatorname{cap}(e)$$

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

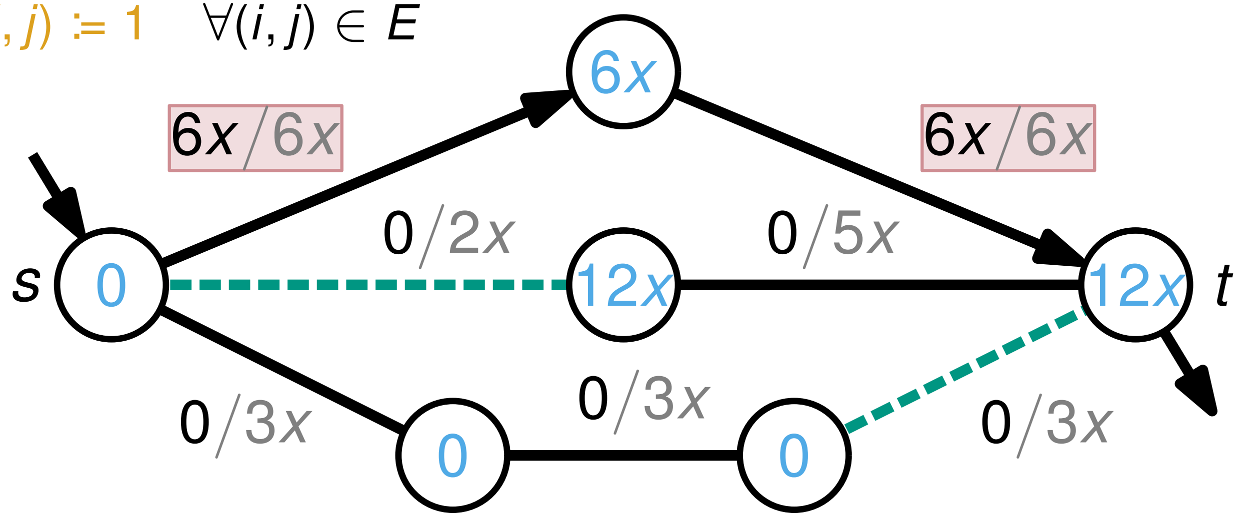
$$\Delta\theta^v(\pi_3) = 3 \cdot 3x = 9x$$

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

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Schalten auf parallelen Pfaden

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



$F = 6x$

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

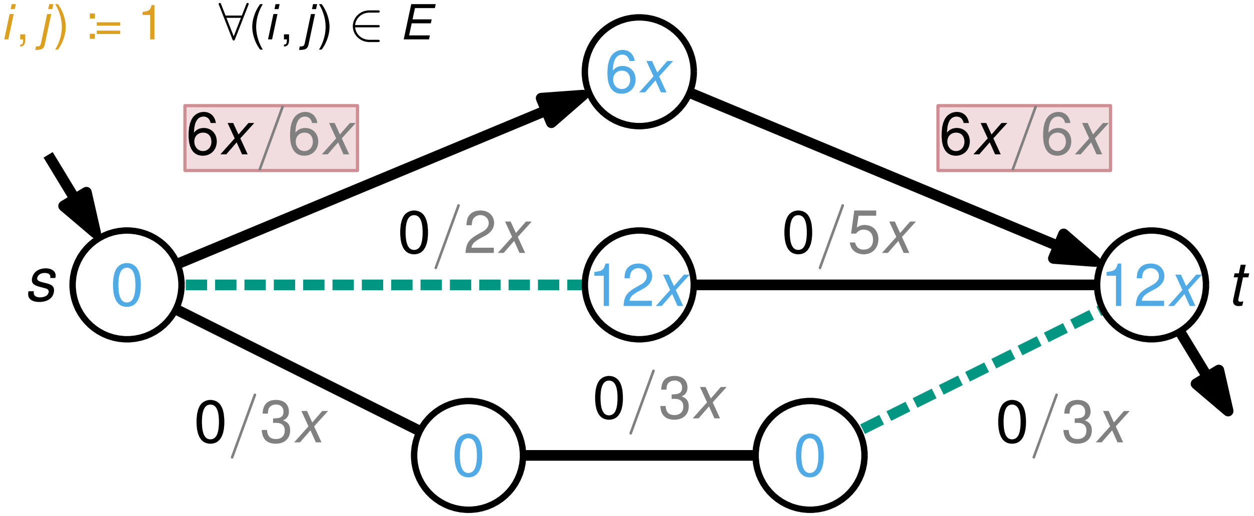
$$\Delta\theta^{v}(\pi_1) = 2 \cdot 6x = 12x$$

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

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

Schalten auf parallelen Pfaden

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



$\Delta\theta^v(\pi) \coloneqq \underbrace{\|\pi\|_b}_{\text{elektrische Distanz}} \cdot \min_{e\in\pi} \mathrm{cap}(e)$

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

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

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

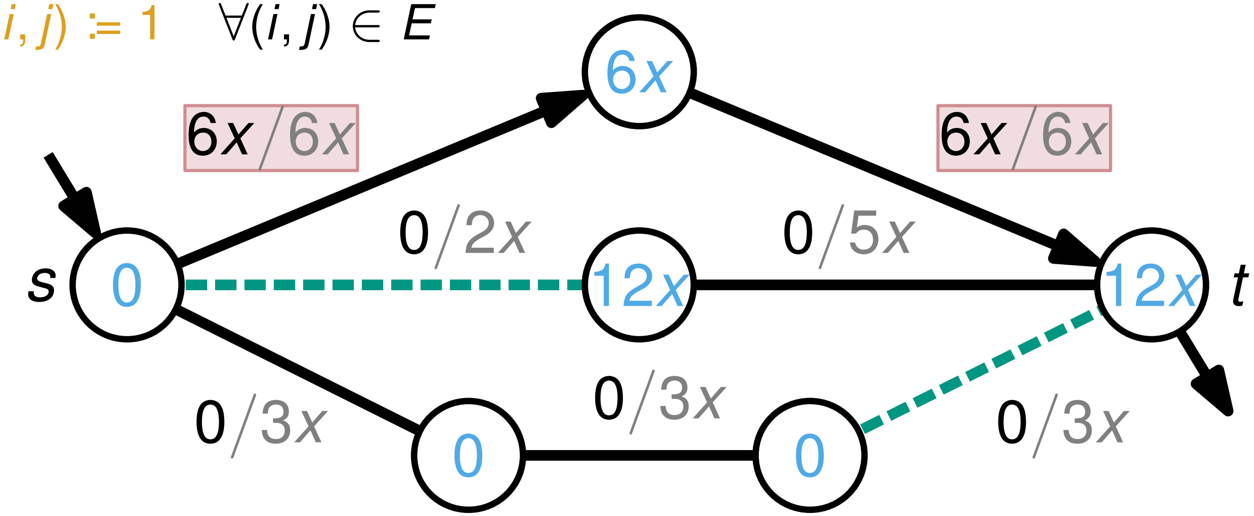
$F = 6x$

Auf parallelen Pfaden kann ein optimales Schalten in Polynomialzeit berechnet werden.

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Schalten auf parallelen Pfaden

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



$\Delta\theta^v(\pi) \coloneqq \underbrace{\|\pi\|_b}_{\text{elektrische Distanz}} \cdot \min_{e\in\pi} \mathrm{cap}(e)$

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

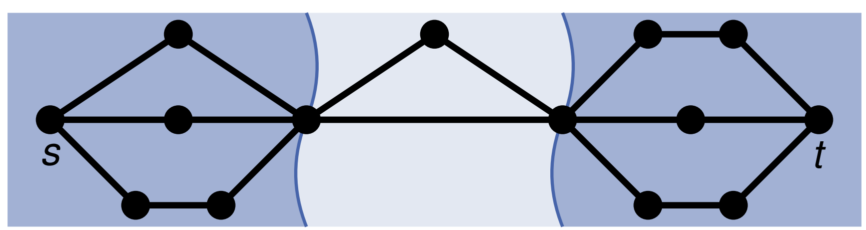
$\Delta\theta^v(\pi_2) = 2 \cdot 2x = 4x$

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

$F = 6x$

Auf parallelen Pfaden kann ein optimales Schalten in Polynomialzeit berechnet werden.

Das selbige gilt für Serienkompositionen von diesen Graphen.



Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Dominating Theta Path (DTP)

[Abschnitt 5; Grastien et al., 2018]

Gegeben seien $u, v \in V$ und ein u-v-Pfad π.

Susceptanznorm:

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

Minimale Kapazität:

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

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Dominating Theta Path (DTP)

[Abschnitt 5; Grastien et al., 2018]

Gegeben seien $u, v \in V$ und ein u-v-Pfad π.

Susceptanznorm:

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

Minimale Kapazität:

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

Spannungswinkeldifferenz von π:

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

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Dominating Theta Path (DTP)

[Abschnitt 5; Grastien et al., 2018]

Gegeben seien $u, v \in V$ und ein u-v-Pfad π.

Susceptanznorm:

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

Minimale Kapazität:

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

Spannungswinkeldifferenz von π:

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

Dominating Theta Path (DTP):

$$\underline{\Delta\theta^{v}}(u, v) := \min\{\Delta\theta^{v}(\pi) \mid \pi \text{ ist ein } u\text{-}v\text{-Pfad}\}$$

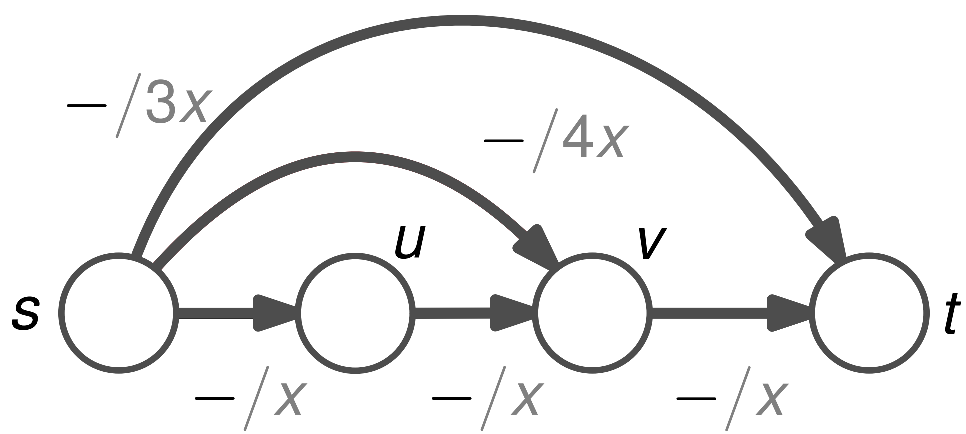
Institut für Theoretische Informatik
Lehrstuhl Algorithmik

DTP Berechnung [Abschnitt 5; Grastien et al., 2018]

Beschreibung:

- Bikriterieller Dijkstra mit Labeln $(\|\pi\|_b, \underline{\mathrm{cap}}(\pi), V_i)$
- maximal $|E|$ Label pro Knoten

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

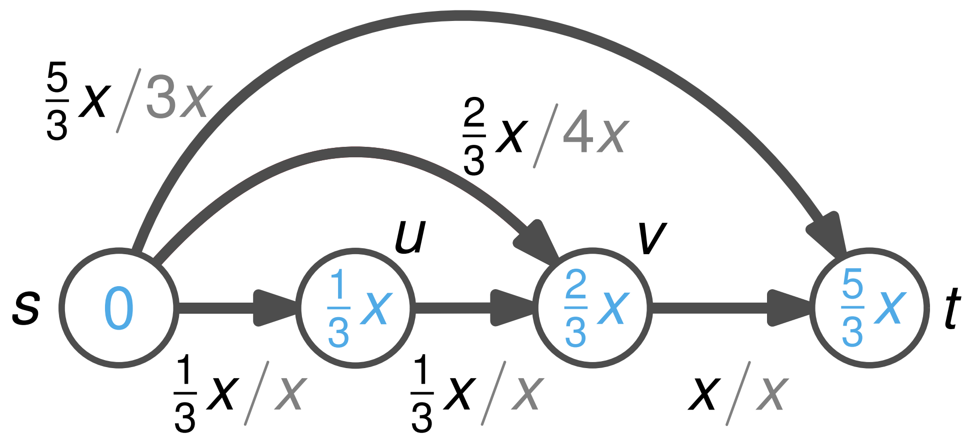


DTP Berechnung [Abschnitt 5; Grastien et al., 2018]

Beschreibung:

- Bikriterieller Dijkstra mit Labeln $(\|\pi\|_b, \mathrm{cap}(\pi), V_i)$
- maximal $|E|$ Label pro Knoten

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



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

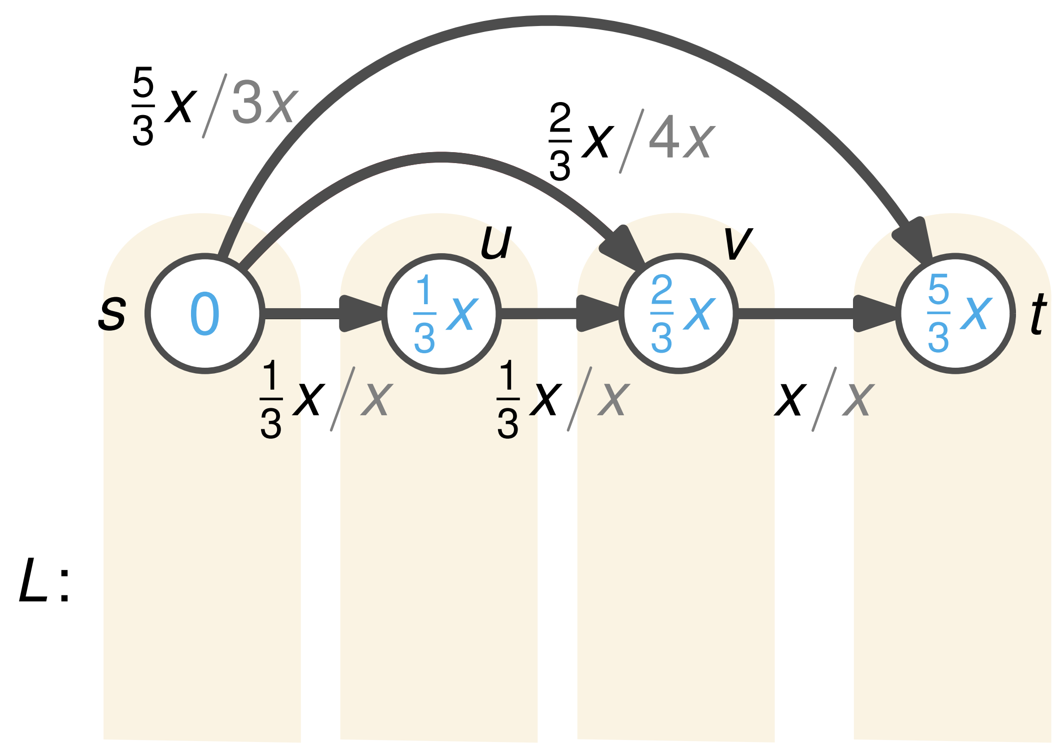
Institut für Theoretische Informatik
Lehrstuhl Algorithmik

DTP Berechnung [Abschnitt 5; Grastien et al., 2018]

Beschreibung:

- Bikriterieller Dijkstra mit Labeln $(\|\pi\|_b, \underline{\mathrm{cap}}(\pi), V_i)$
- maximal $|E|$ Label pro Knoten

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



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

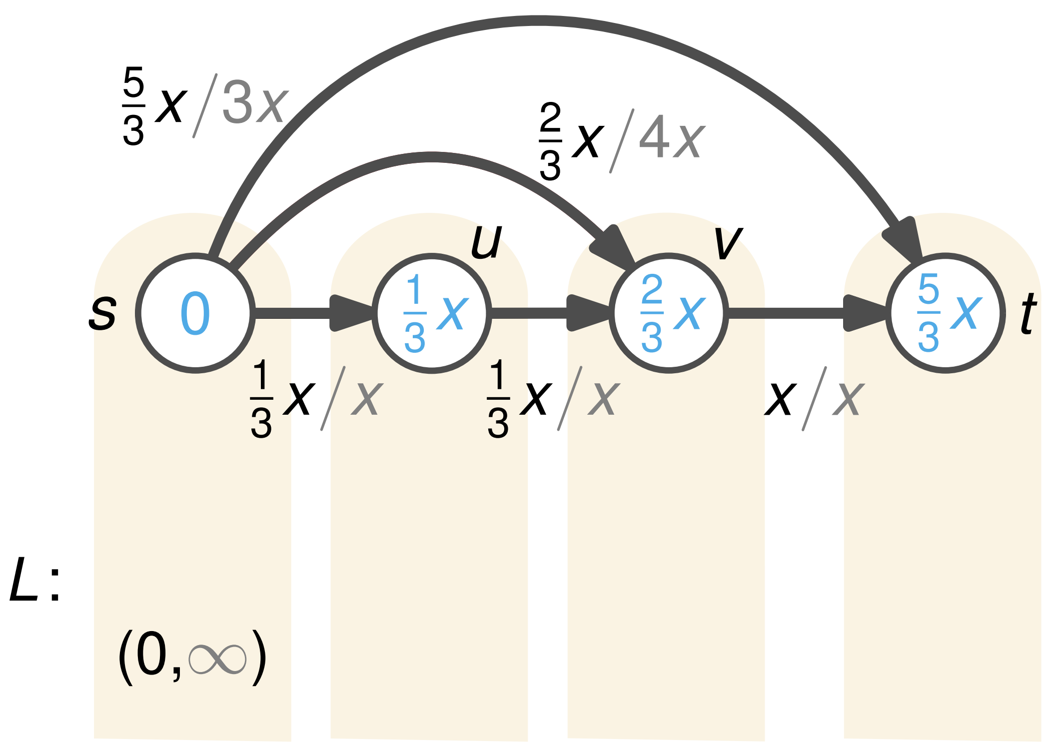
Institut für Theoretische Informatik
Lehrstuhl Algorithmik

DTP Berechnung [Abschnitt 5; Grastien et al., 2018]

Beschreibung:

- Bikriterieller Dijkstra mit Labeln $(\|\pi\|_b, \underline{\mathrm{cap}}(\pi), V_i)$
- maximal $|E|$ Label pro Knoten

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



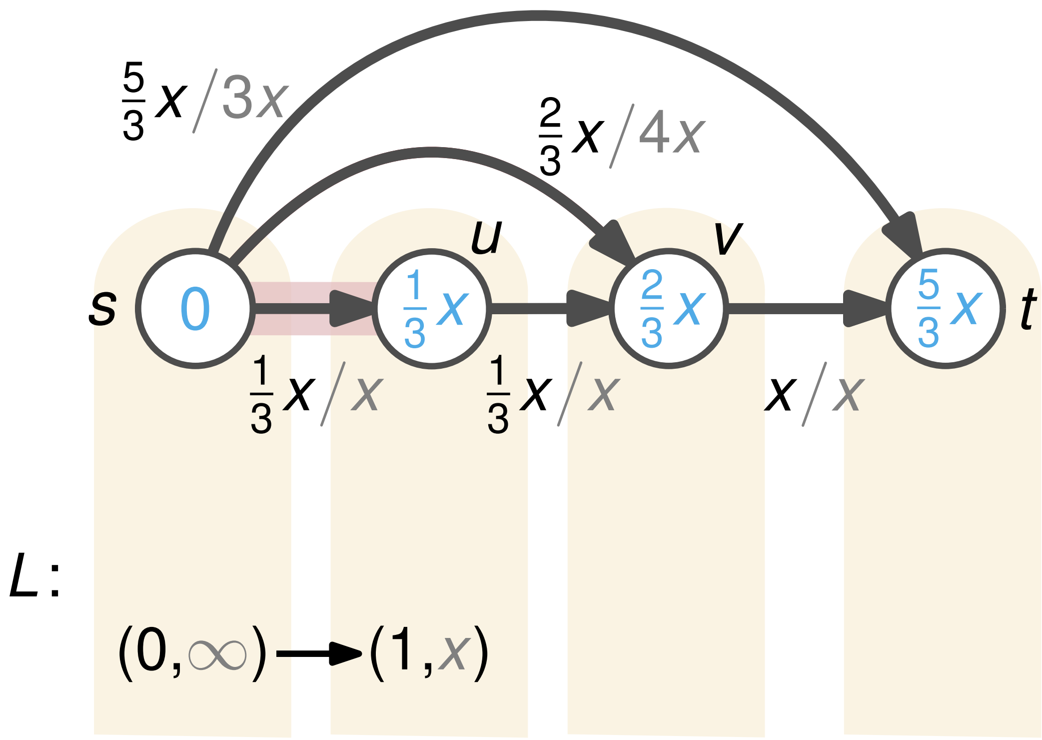
$\mathrm{OPT}_{\mathrm{MPFP}} = \frac{8}{3}x$

DTP Berechnung [Abschnitt 5; Grastien et al., 2018]

Beschreibung:

- Bikriterieller Dijkstra mit Labeln $(\|\pi\|_b, \underline{\mathrm{cap}(\pi)}, V_i)$
- maximal $|E|$ Label pro Knoten

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

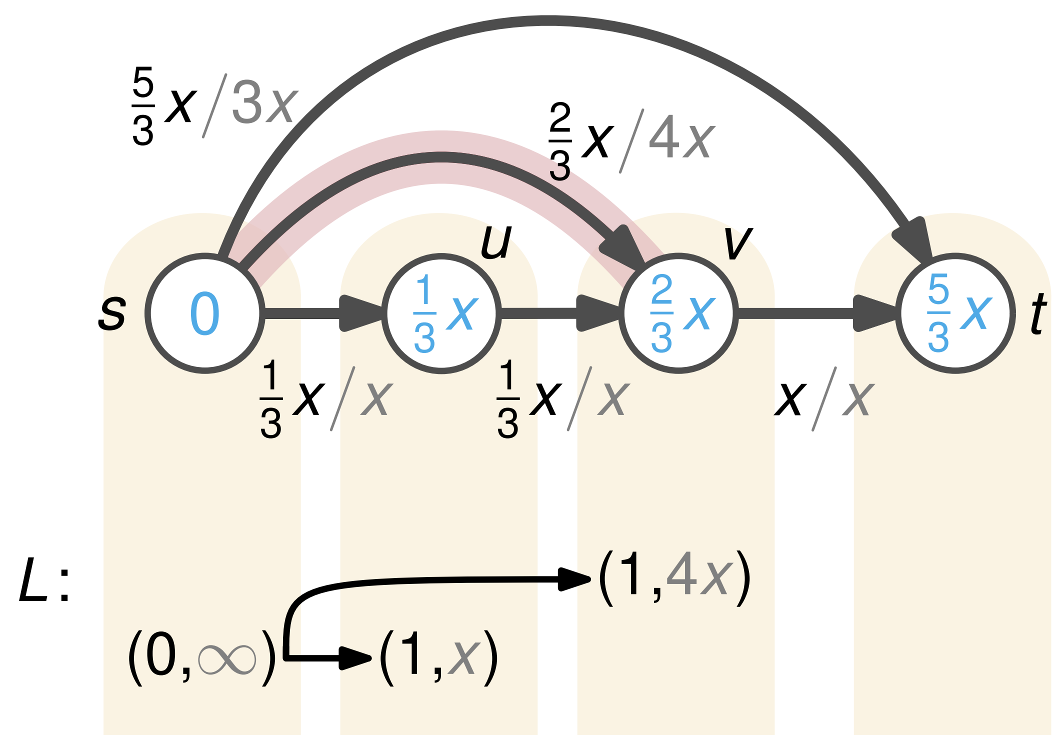


$\mathrm{OPT}_{\mathrm{MPFP}} = \frac{8}{3}x$

Beschreibung:

- Bikriterieller Dijkstra mit Labeln $(\|\pi\|_b, \mathrm{cap}(\pi), V_i)$
- maximal $|E|$ Label pro Knoten

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

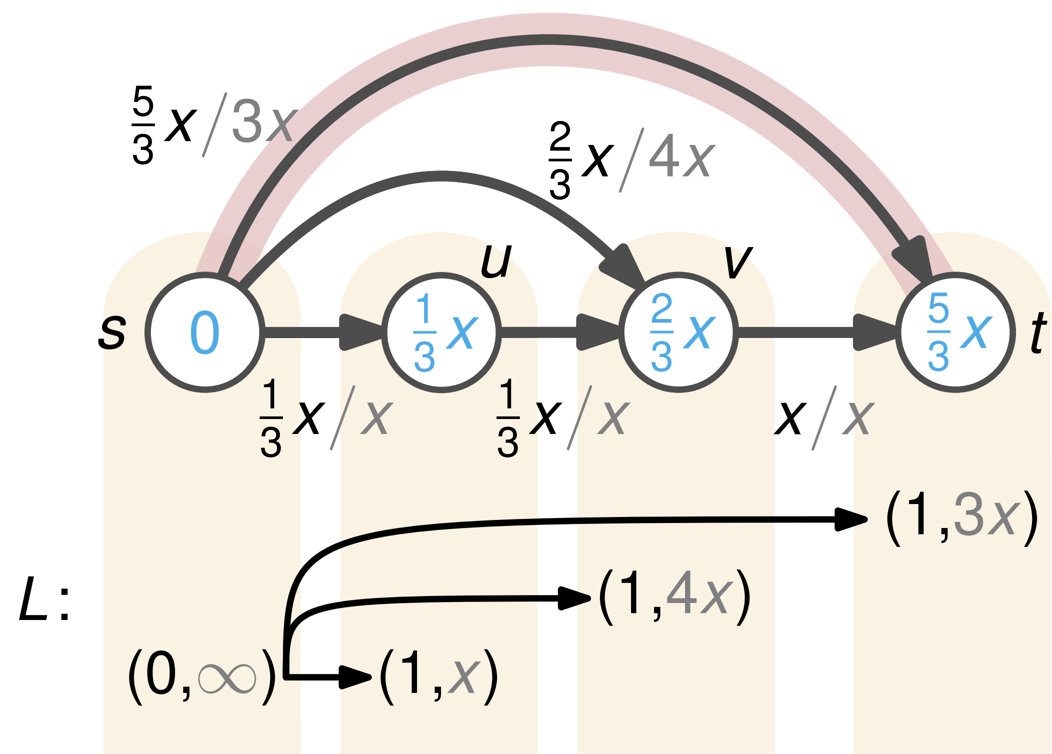


$\mathrm{OPT}_{\mathrm{MPFP}} = \frac{8}{3}x$

Beschreibung:

- Bikriterieller Dijkstra mit Labeln $(\|\pi\|_b, \underline{\mathrm{cap}}(\pi), V_i)$
- maximal $|E|$ Label pro Knoten

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



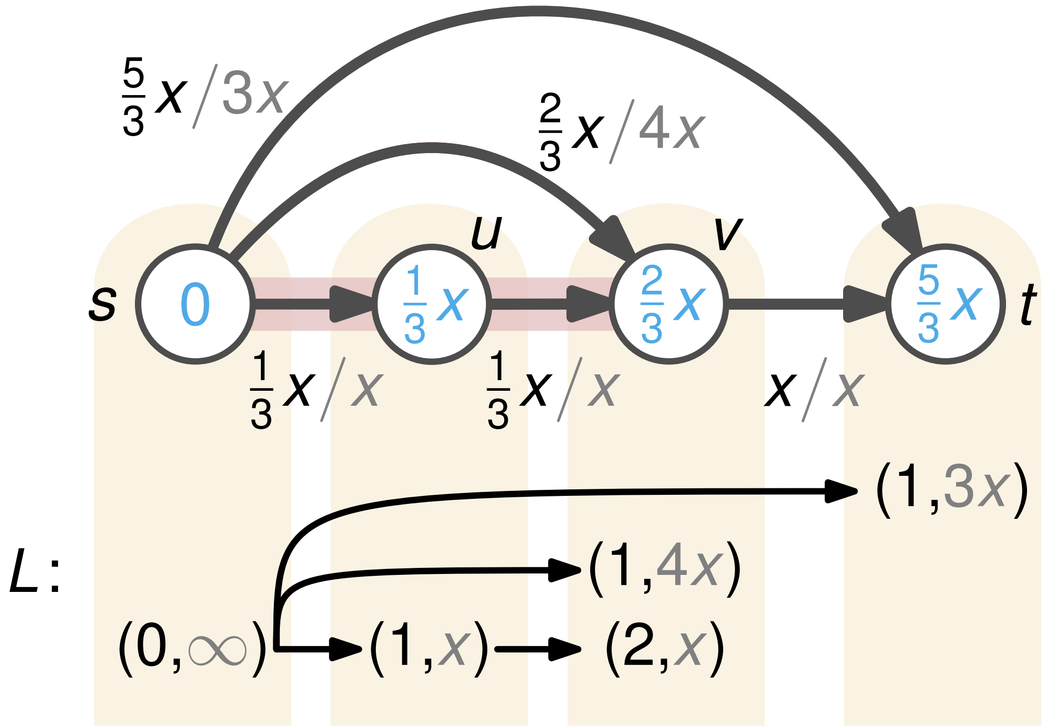
$\mathrm{OPT}_{\mathrm{MPFP}} = \frac{8}{3}x$

DTP Berechnung [Abschnitt 5; Grastien et al., 2018]

Beschreibung:

- Bikriterieller Dijkstra mit Labeln $(\|\pi\|_b, \underline{\mathrm{cap}(\pi)}, V_i)$
- maximal $|E|$ Label pro Knoten

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



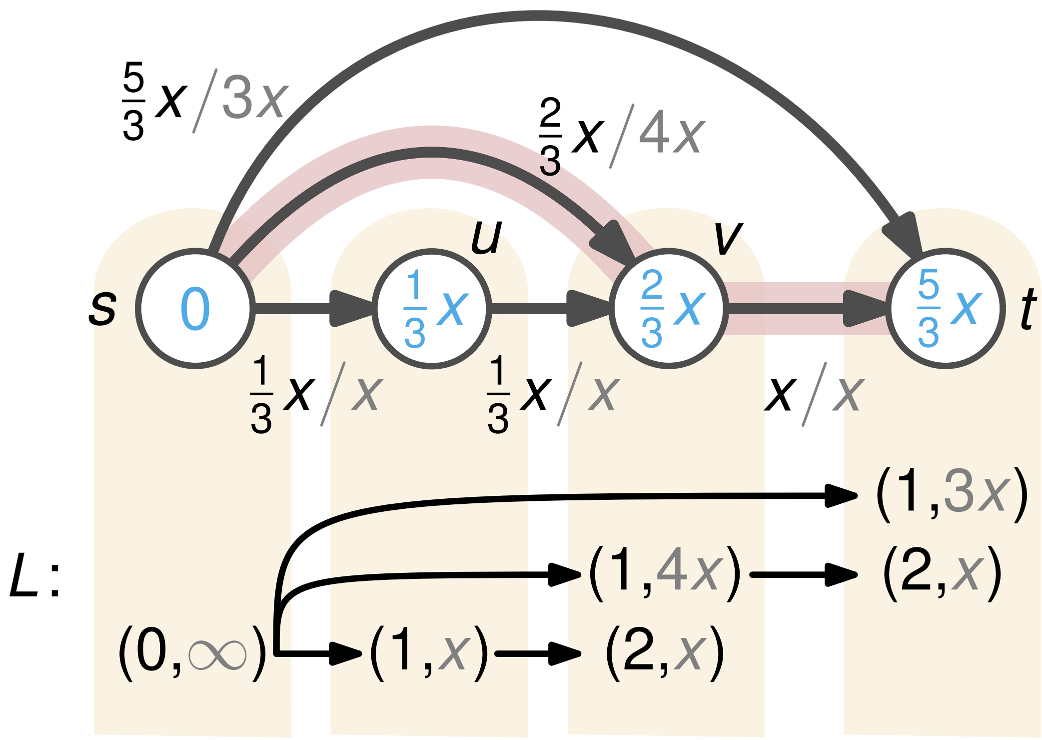
$\mathrm{OPT}_{\mathrm{MPFP}} = \frac{8}{3}x$

DTP Berechnung [Abschnitt 5; Grastien et al., 2018]

Beschreibung:

- Bikriterieller Dijkstra mit Labeln $(\|\pi\|_b, \mathrm{cap}(\pi), V_i)$
- maximal $|E|$ Label pro Knoten

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



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

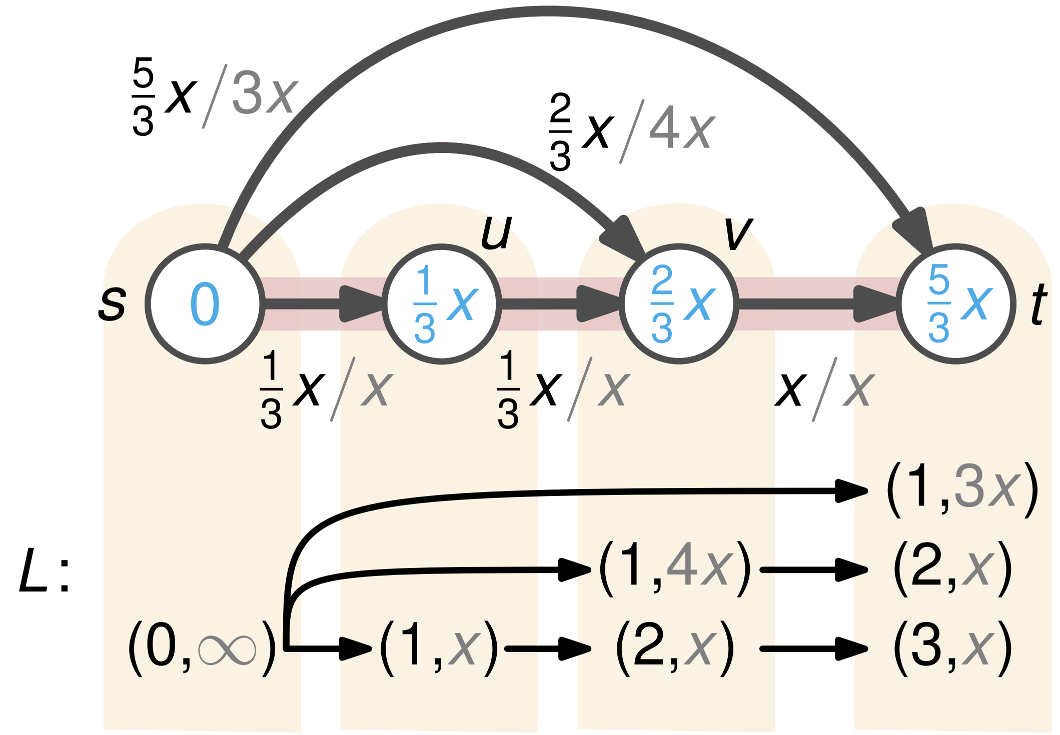
Institut für Theoretische Informatik
Lehrstuhl Algorithmik

DTP Berechnung [Abschnitt 5; Grastien et al., 2018]

Beschreibung:

- Bikriterieller Dijkstra mit Labeln $(\|\pi\|_b, \underline{\mathrm{cap}}(\pi), V_i)$
- maximal $|E|$ Label pro Knoten

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



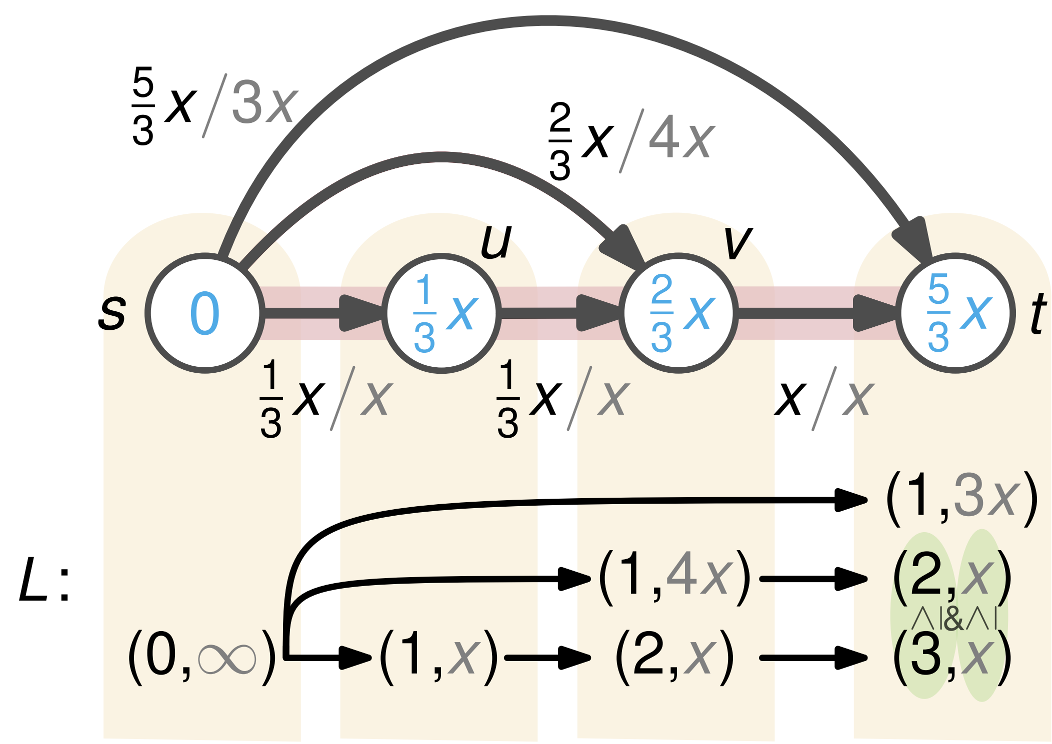
$\mathrm{OPT}_{\mathrm{MPFP}} = \frac{8}{3}x$

DTP Berechnung [Abschnitt 5; Grastien et al., 2018]

Beschreibung:

- Bikriterieller Dijkstra mit Labeln $(\|\pi\|_b, \underline{\mathrm{cap}}(\pi), V_i)$
- maximal $|E|$ Label pro Knoten

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

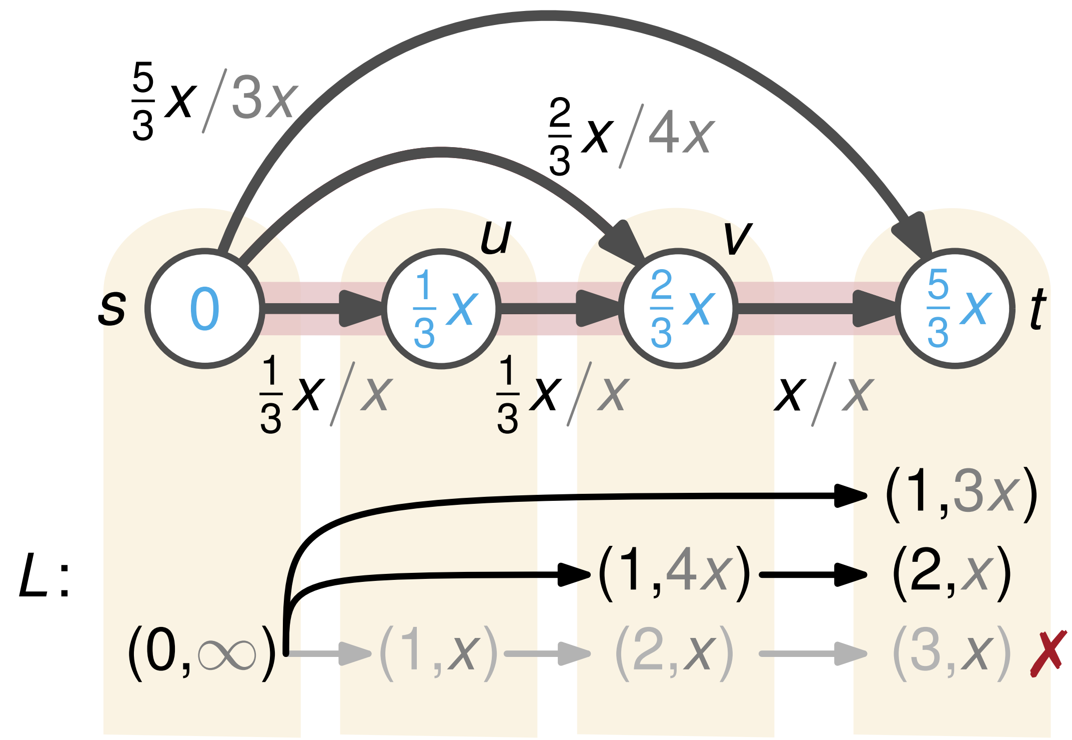


$\mathrm{OPT}_{\mathrm{MPFP}} = \frac{8}{3}x$

Beschreibung:

- Bikriterieller Dijkstra mit Labeln $(\|\pi\|_b, \underline{\mathrm{cap}(\pi)}, V_i)$
- maximal $|E|$ Label pro Knoten

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



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

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

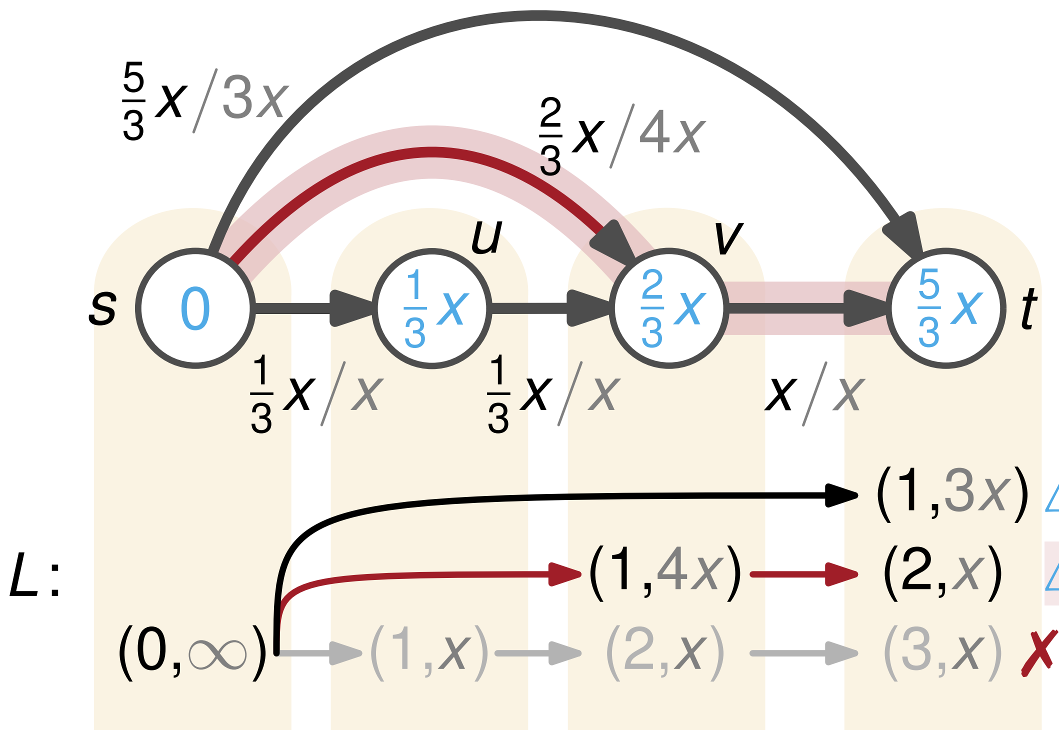
DTP Berechnung [Abschnitt 5; Grastien et al., 2018]

Beschreibung:

- Bikriterieller Dijkstra mit Labeln $(\|\pi\|_b, \underline{\text{cap}(\pi)}, V_i)$
- maximal $|E|$ Label pro Knoten

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

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



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

$\geq$

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

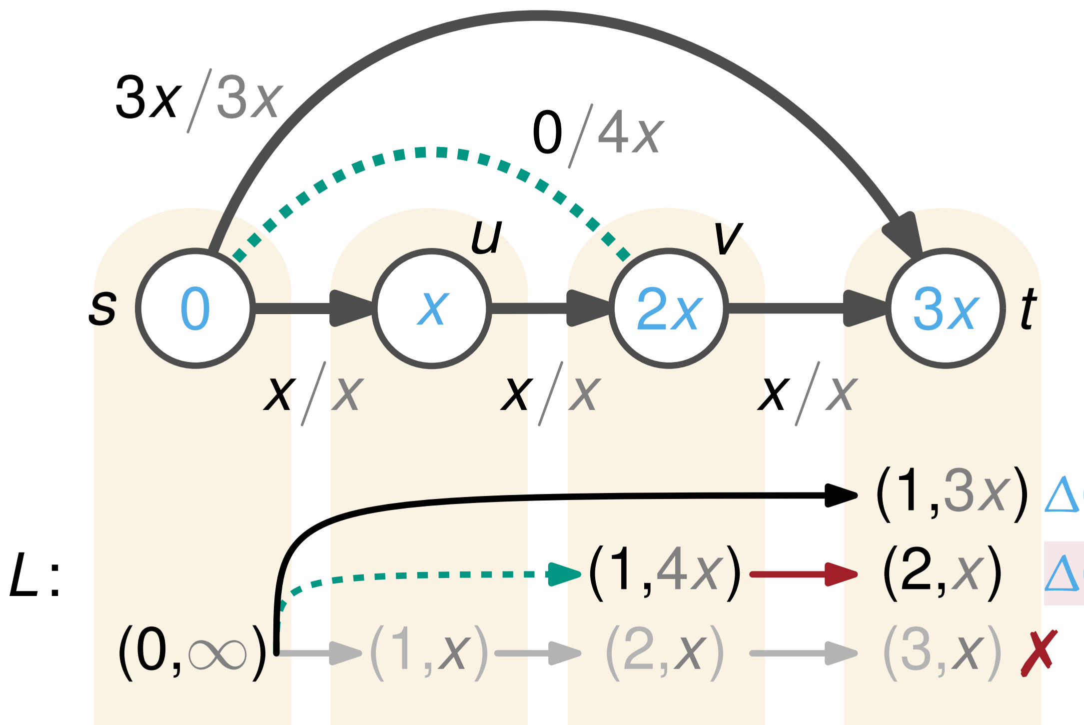
Institut für Theoretische Informatik
Lehrstuhl Algorithmik

DTP Berechnung [Abschnitt 5; Grastien et al., 2018]

Beschreibung:

- Bikriterieller Dijkstra mit Labeln $(\|\pi\|_b, \mathrm{cap}(\pi), V_i)$
- maximal $|E|$ Label pro Knoten

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



$\mathrm{OPT}_{\mathrm{MPFP}} = 4x = \mathrm{OPT}_{\mathrm{MTSF}}$

$\Delta\theta^{v}(s,t) = 1 \cdot 3x = 3x$

$\vee\!|$

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

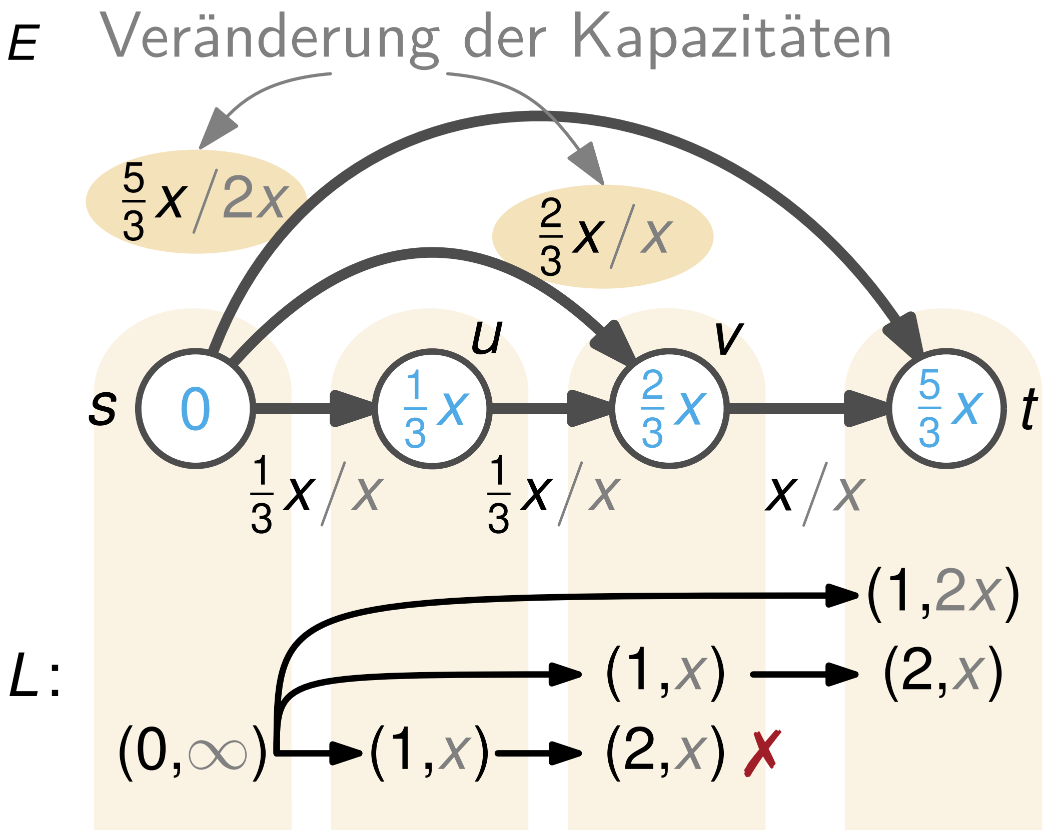
Institut für Theoretische Informatik
Lehrstuhl Algorithmik

DTP Berechnung [Abschnitt 5; Grastien et al., 2018]

Beschreibung:

- Bikriterieller Dijkstra mit Labeln $(\|\pi\|_b, \mathrm{cap}(\pi), V_i)$
- maximal $|E|$ Label pro Knoten

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



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

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

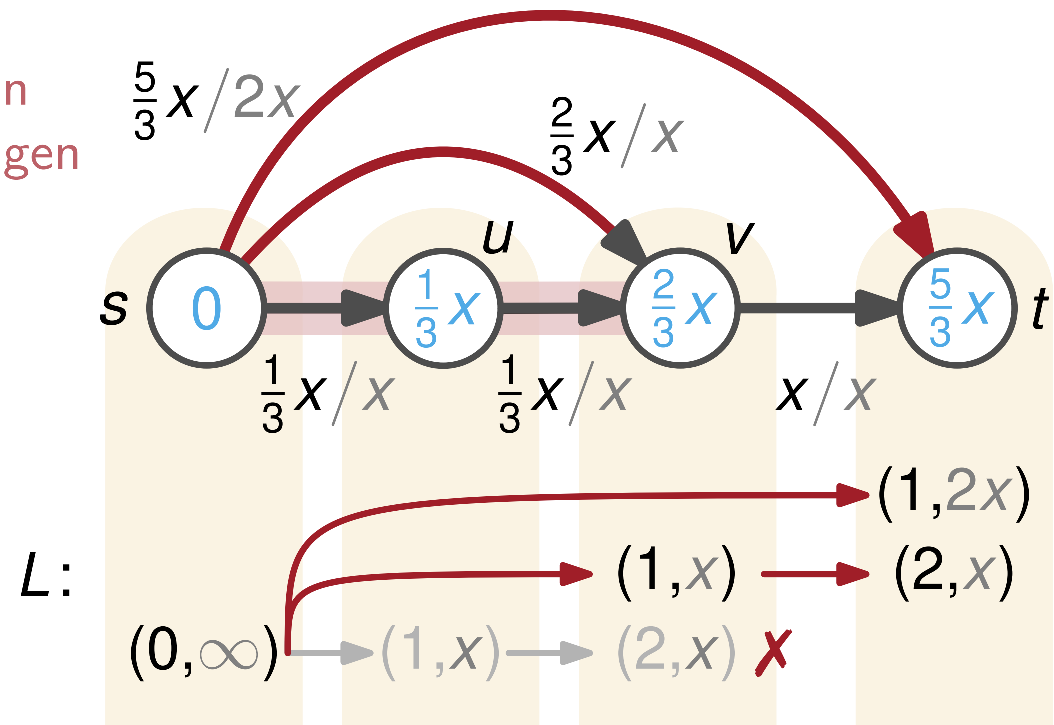
DTP Berechnung [Abschnitt 5; Grastien et al., 2018]

Beschreibung:

- Bikriterieller Dijkstra mit Labeln $(\|\pi\|_b, \mathrm{cap}(\pi), V_i)$
- maximal $|E|$ Label pro Knoten

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

- DTPs von s müssen keinen Baum erzeugen



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

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

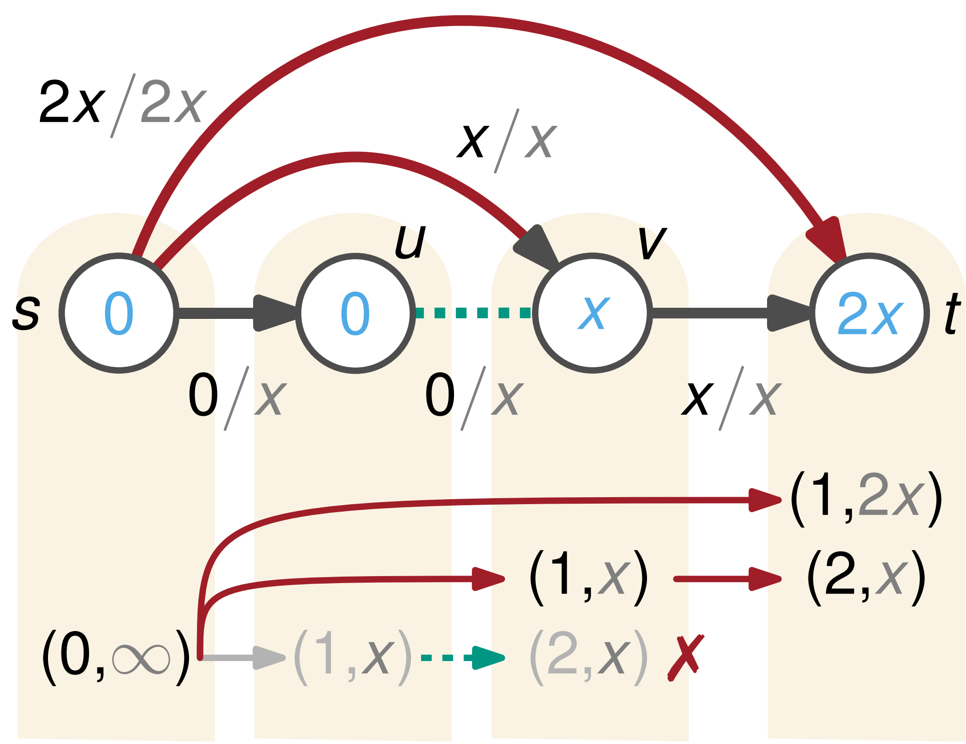
DTP Berechnung [Abschnitt 5; Grastien et al., 2018]

Beschreibung:

- Bikriterieller Dijkstra mit Labeln $(\|\pi\|_b, \underline{\mathrm{cap}(\pi)}, V_i)$
- maximal $|E|$ Label pro Knoten

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

- DTPs von s müssen keinen Baum erzeugen
- Optimal platzierte Schalter müssen nicht auf dem DTP liegen, wenn die zugrundeliegende Struktur <u>kein</u> Penrose-Minoren-freier Graph ist



$\mathrm{OPT}_{\mathrm{MPFP}} = 3x$
$= \mathrm{OPT}_{\mathrm{MTSFP}}$

[S.345; Grastien et al., 2018]

Karlsruher Institut für Technologie

Daten: Ein Netzwerk $\mathcal{N} = (G, V_G, V_D, \text{cap}, b)$.

Ergebnis: $\pi_{\text{DTP}}(s,t)$, $\underline{\Delta\theta^V(s,t)}$, und $D(v)$ mit $v \in V$.

$D(u) \coloneqq L(u) \coloneqq \emptyset;\ \forall u \in V;$ ▷ Initialiesierung
$Q \coloneqq \emptyset;$
$L(s) \coloneqq \{(0,\infty)\}$ ▷ Spezielles Label für die Quelle s
Q.insert$\big((0,\infty), s, \text{key}((0,\infty))\big)$;
} $\mathcal{O}(|V|)$

while $Q \neq \emptyset$ **do** ▷ Besuche alle Knoten — $|V|\cdot|E|$
  $(\ell, u, \text{key}) \coloneqq Q$.deleteMin(); — $|V|\cdot|E|\cdot\mathcal{O}(\log|V|)$
  $D(u) \coloneqq D(u) \cup \{\ell\}$;
  for $\forall\{u,v\} \in \overleftrightarrow{E}$ **do** ▷ Überprüfe adjazente Knoten — $|E|^2$
    $\underline{\text{cap}}(\pi(s,u,v)) \coloneqq \min(\ell[1], \text{cap}(u,v))$; — $|E|^2\cdot\mathcal{O}(1)$
    $\ell_{\text{new}}(v) \coloneqq \left(\ell[0] + \frac{1}{b(u,v)}, \underline{\text{cap}}(\pi(s,u,v))\right)$; — $|E|^2\cdot\mathcal{O}(1)$
    If 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**
        parent$(\ell_{\text{new}}(v)) \coloneqq$ parent$(\ell_{\text{new}}(v)) \cup \{\ell\}$; — $|E|^2\cdot\mathcal{O}(1)$
      else if not $L(v)$ **dominates** $\ell_{\text{new}}(v)$ **then**
        $L(v)$.deleteDominatedLabels$(\ell_{\text{new}}(v))$; — $|E|^2\cdot\mathcal{O}(|E|)$
        Q.deleteDominatedLabels$(\ell_{\text{new}}(v), v)$; — $|E|^2\cdot\mathcal{O}(|E|)$
        $L(v)$.insert$(\ell_{\text{new}}(v))$; — $|E|^2\cdot\mathcal{O}(1)$
        Q.insert$(\ell_{\text{new}}(v), v, \text{key}(\ell_{\text{new}}(v)))$; — $|E|^2\cdot\mathcal{O}(1)$
        parent$(\ell_{\text{new}}(v)) \coloneqq \{\ell\}$; — $|E|^2\cdot\mathcal{O}(1)$
      end
    end
  end
end
return $\left(\pi_{\text{DTP}}(s,t) \coloneqq \text{getPaths}(s,t),\ \underline{\Delta\theta^V}(s,t) \coloneqq \min_{\ell\in D(t)}\{\ell[0]\cdot\ell[1]\},\ D(\cdot)\right)$ ▷ Baue Pfad von den Eltern

iteriere über alle Label im Beutel, $|E|$ Tests pro Beutel

Jeder Knoten besitzt maximal $|E|$ Label

Königsberg-Brücken-Problem

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

Fibonacci-heap $\mathcal{Q}$

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

Laufzeitanalyse für den DTP Algorithmus

[S.345; Grastien et al., 2018]

Karlsruher Institut für Technologie

Daten: Ein Netzwerk $\mathcal{N} = (G, V_G, V_D, \text{cap}, b)$.

Ergebnis: $\pi_{\text{DTP}}(s,t)$, $\Delta\theta^V(s,t)$, und $D(v)$ mit $v \in V$.

$D(u) \coloneqq L(u) \coloneqq \emptyset$; $\forall u \in V$; ▷ Initialiesierung
$Q \coloneqq \emptyset$;
$L(s) \coloneqq \{(0,\infty)\}$ ▷ Spezielles Label für die Quelle s
Q.insert$((0,\infty), s, \text{key}((0,\infty)))$;

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

while $Q \neq \emptyset$ **do** ▷ Besuche alle Knoten — $|V| \cdot |E|$
  $(\ell, u, \text{key}) \coloneqq Q$.deleteMin(); — $|V| \cdot |E| \cdot \mathcal{O}(\log|V|)$
  $D(u) \coloneqq D(u) \cup \{\ell\}$;
  for $\forall \{u,v\} \in \overleftrightarrow{E}$ **do** ▷ Überprüfe adjazente Knoten — $|E|^2$

Auf allgemeinen Graphen läuft der DTP Algorithmus in $\mathcal{O}\left(2^{|V|}|V| \cdot |E|^3\right)$ Zeit.

      If $\ell_{\text{new}}(v) \in L(v)$ **then**
        parent$(\ell_{\text{new}}(v)) \coloneqq$ parent$(\ell_{\text{new}}(v)) \cup \{\ell\}$; — $|E|^2 \cdot \mathcal{O}(1)$
      else if not $L(v)$ **dominates** $\ell_{\text{new}}(v)$ **then**
        $L(v)$.deleteDominatedLabels$(\ell_{\text{new}}(v))$; — $|E|^2 \cdot \mathcal{O}(|E|)$
        Q.deleteDominatedLabels$(\ell_{\text{new}}(v), v)$; — $|E|^2 \cdot \mathcal{O}(|E|)$
        $L(v)$.insert$(\ell_{\text{new}}(v))$; — $|E|^2 \cdot \mathcal{O}(1)$
        Q.insert$(\ell_{\text{new}}(v), v, \text{key}(\ell_{\text{new}}(v)))$; — $|E|^2 \cdot \mathcal{O}(1)$
        parent$(\ell_{\text{new}}(v)) \coloneqq \{\ell\}$; — $|E|^2 \cdot \mathcal{O}(1)$
      end → iteriere über alle Label im Beutel, $|E|$ Tests pro Beutel
    end
  end
end
return $\pi_{\text{DTP}}(s,t) \coloneqq$ getPaths(s,t), ▷ Baue Pfad von den Eltern
  $\Delta\theta^V(s,t) \coloneqq \min_{\ell \in D(t)}\{\ell[0] \cdot \ell[1]\}$,
  $D(\cdot)$

Jeder Knoten besitzt m… Beutel

Königsberg-Brücken-P…

$\sum_{v \in V} \deg(v) = 2$…

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

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Laufzeitanalyse für den DTP Algorithmus

[S.345; Grastien et al., 2018]

Karlsruher Institut für Technologie

Daten: Ein Netzwerk $\mathcal{N} = (G, V_G, V_D, \text{cap}, b)$.

Ergebnis: $\pi_{\text{DTP}}(s,t)$, $\Delta\theta^V(s,t)$, und $D(v)$ mit $v \in V$.

```
D(u) := L(u) := ∅; ∀u ∈ V;                       ▷ Initialiesierung      }
Q := ∅;                                                                    }  O(|V|)
L(s) := {(0, ∞)}                       ▷ Spezielles Label für die Quelle s }
Q.insert((0, ∞), s, key((0, ∞)));                                          }
while Q ≠ ∅ do                                 ▷ Besuche alle Knoten        |V|·|E|
    (ℓ, u, key) := Q.deleteMin();                                          |V|·|E|·O(log|V|)
    D(u) := D(u) ∪ {ℓ};
            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),          ▷ Baue Pfad von den Eltern
         Δθ^V(s,t) := min_{ℓ∈D(t)} {ℓ[0] · ℓ[1]},
         D(·) )
```

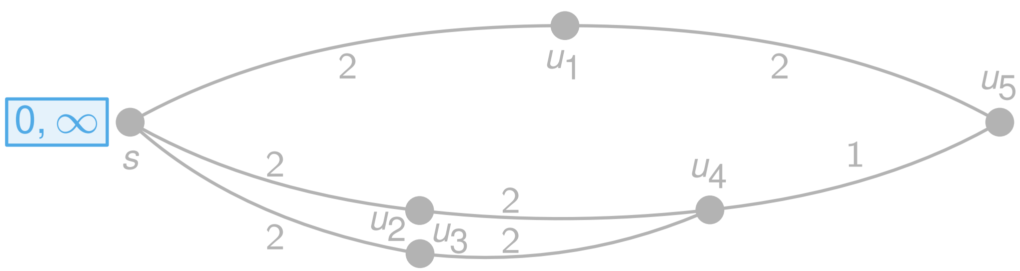
Jeder Knoten besitzt … | Königsberg-Brück… | $\sum_{v \in V} \deg(v)$

→ iteriere über alle Label im Beutel, $|E|$ Tests pro Beutel

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

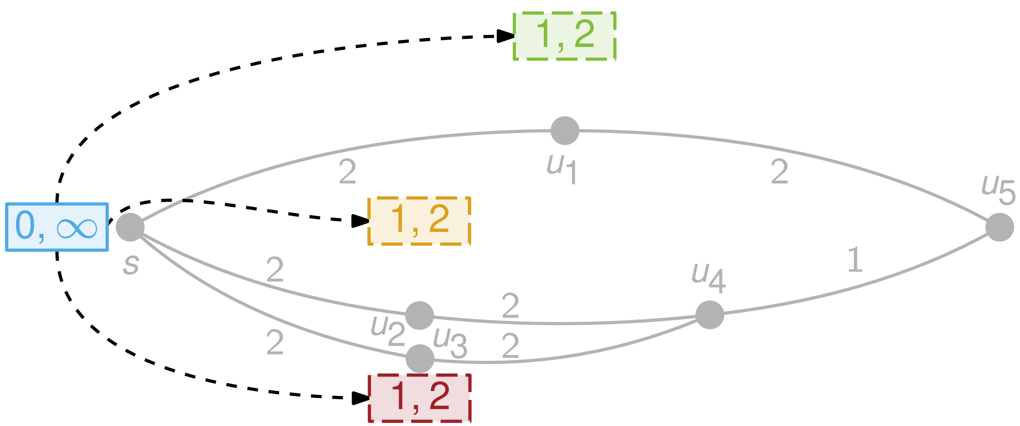
Auf allgemeinen Graphen existiert ein DTP Algorithmus, der in Polynomialzeit läuft und einen DTP zwischen jedem Paar von Knoten u und v berechnet.

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



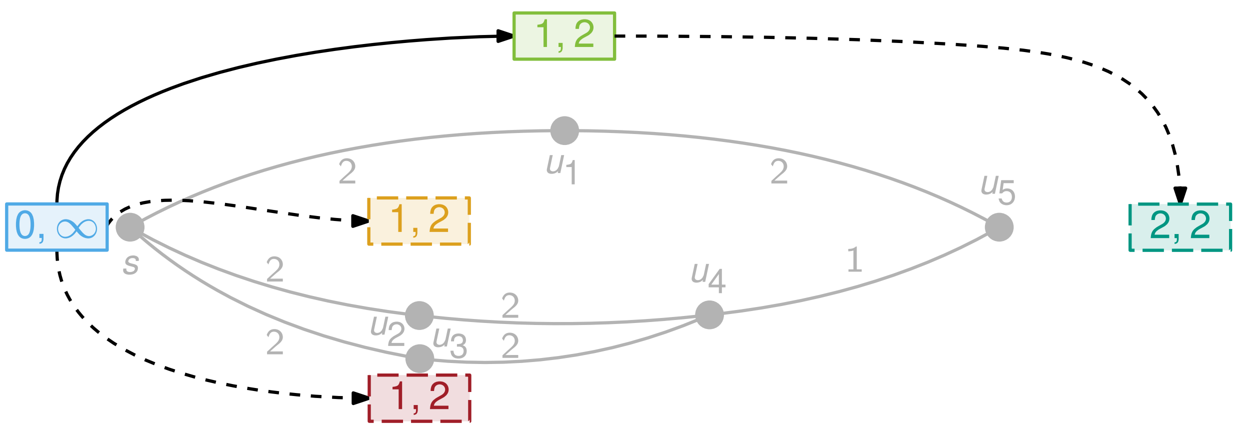
Institut für Theoretische Informatik
Lehrstuhl Algorithmik

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



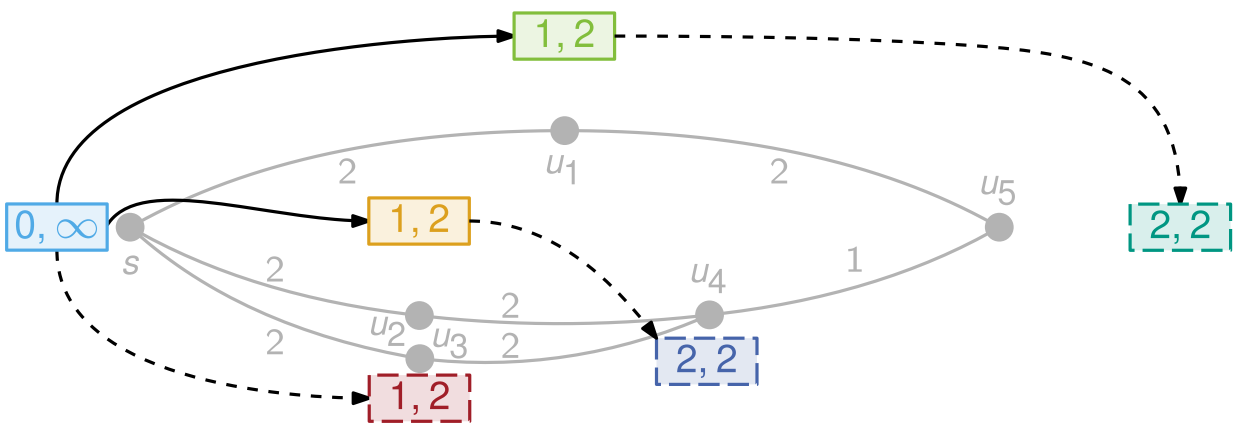
Institut für Theoretische Informatik
Lehrstuhl Algorithmik

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

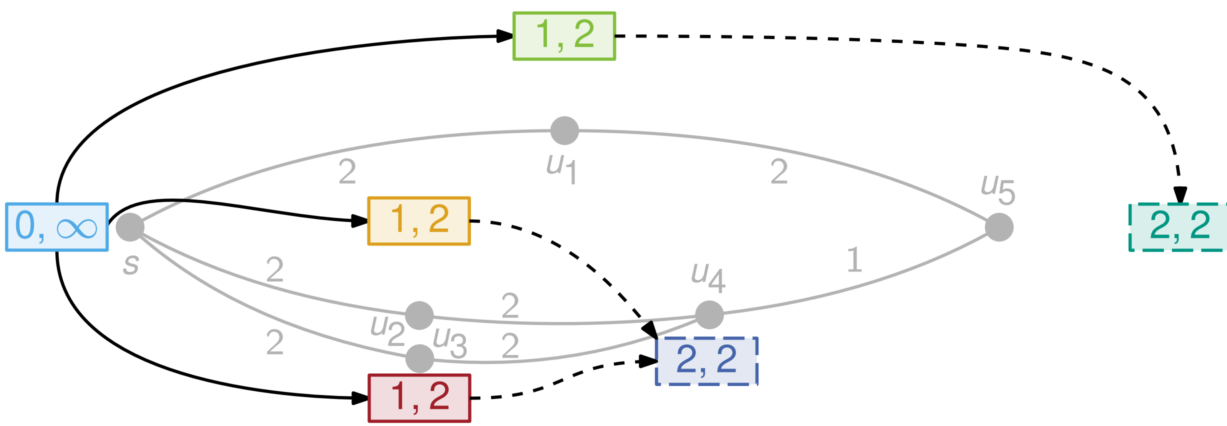


Institut für Theoretische Informatik
Lehrstuhl Algorithmik

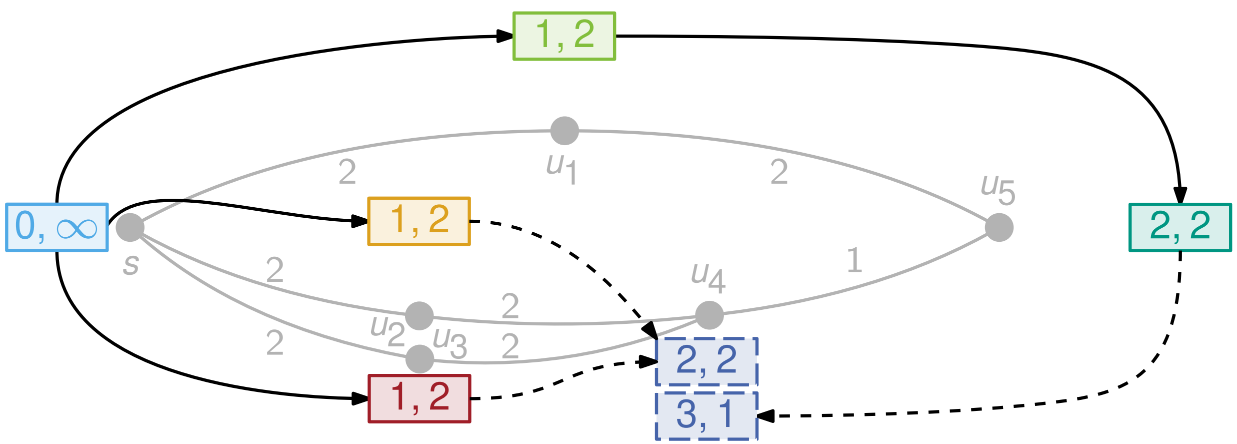
isReachable [S.354; Grastien et al., 2018]



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

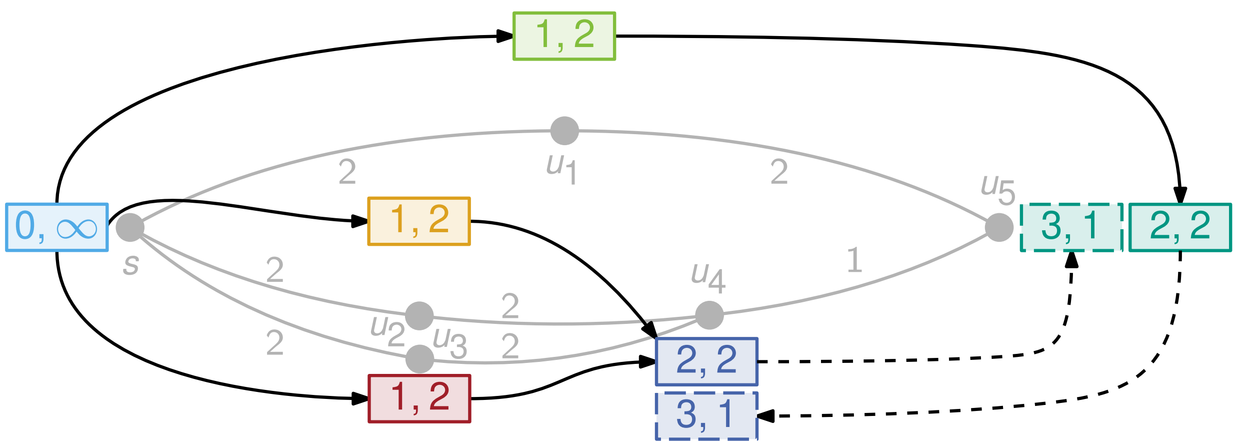


isReachable [S.354; Grastien et al., 2018]

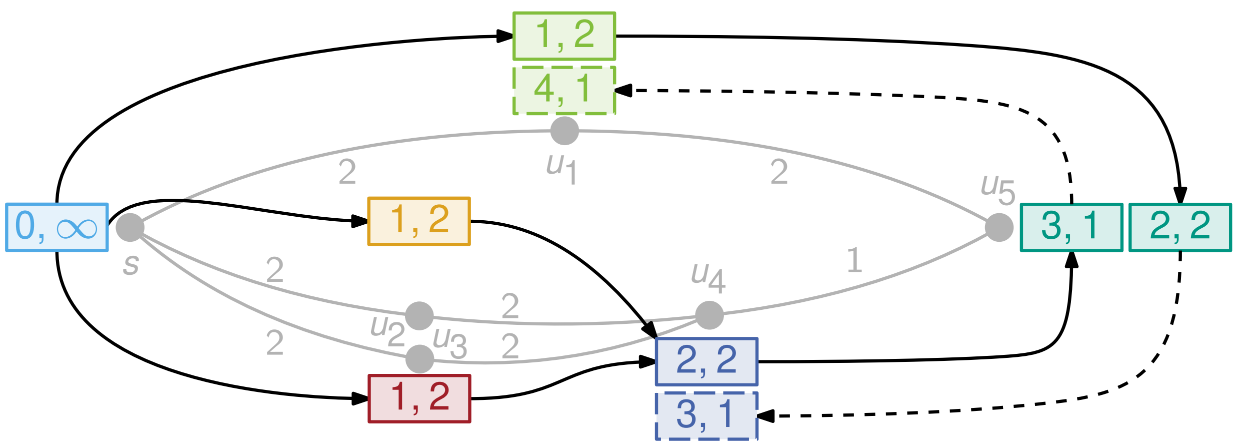


Institut für Theoretische Informatik
Lehrstuhl Algorithmik

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

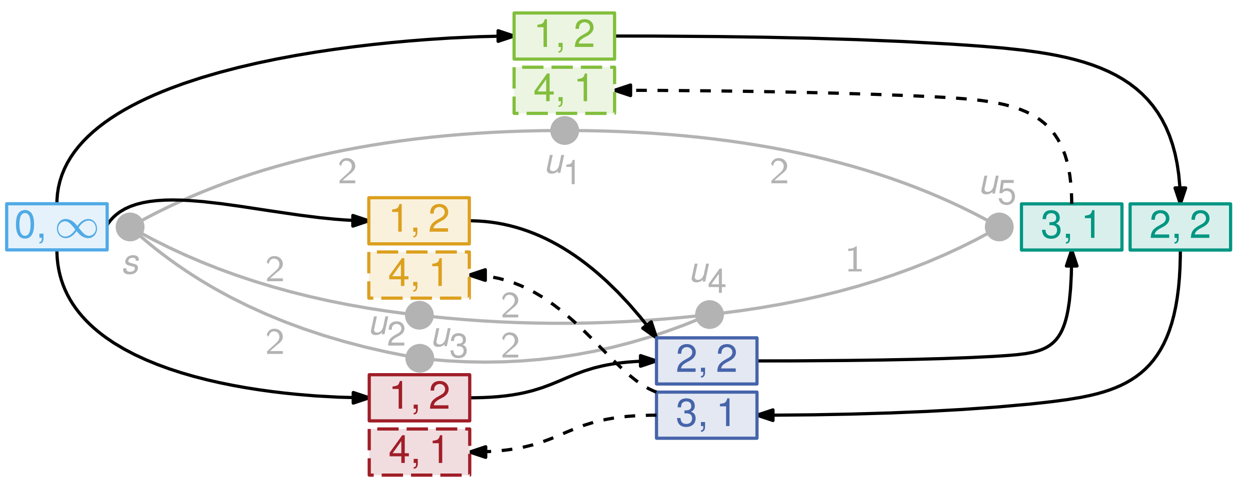


`isReachable` [S.354; Grastien et al., 2018]



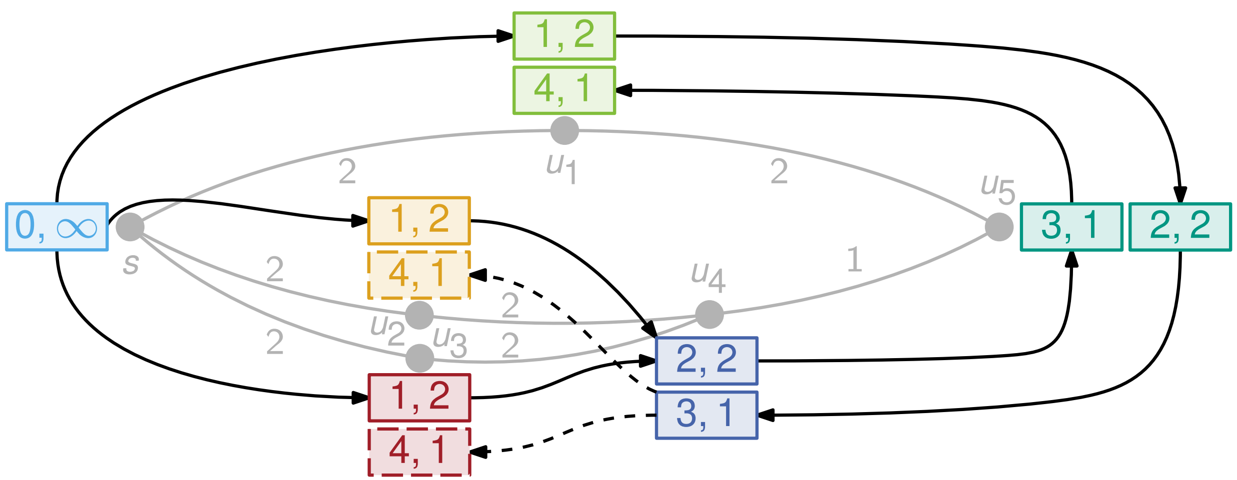
Institut für Theoretische Informatik
Lehrstuhl Algorithmik

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



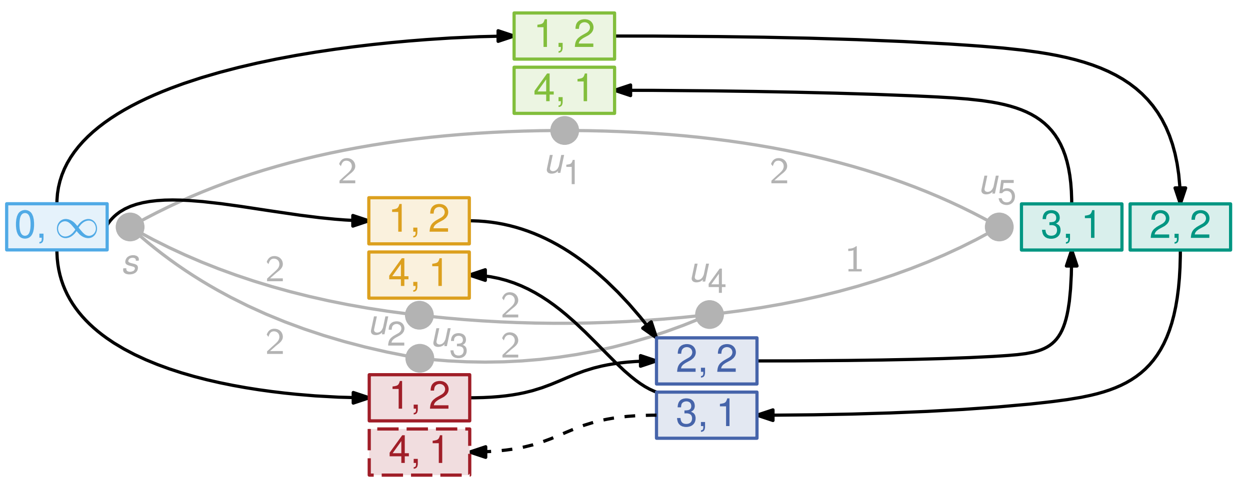
Institut für Theoretische Informatik
Lehrstuhl Algorithmik

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

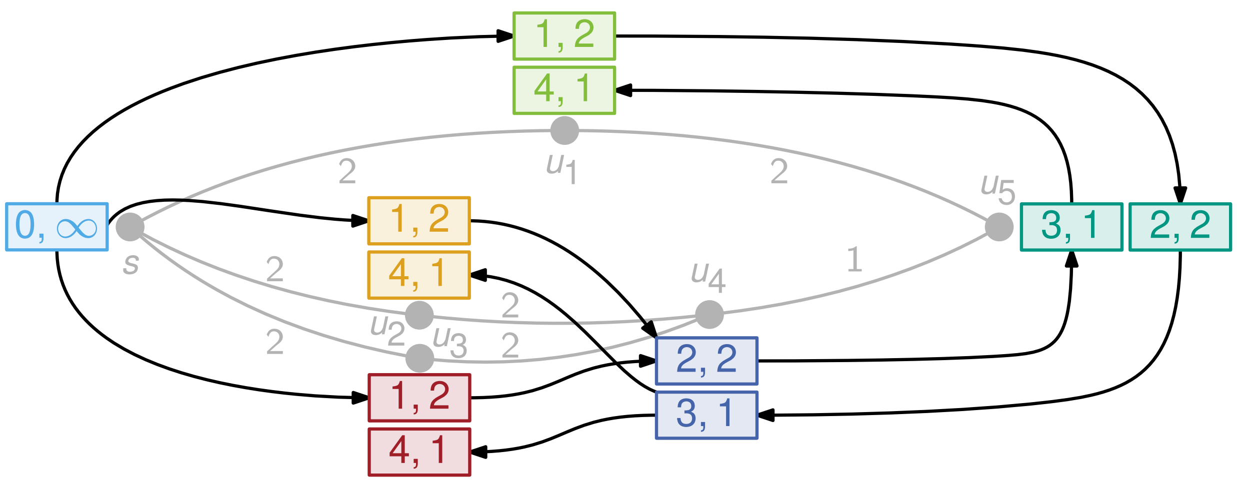


Institut für Theoretische Informatik
Lehrstuhl Algorithmik

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

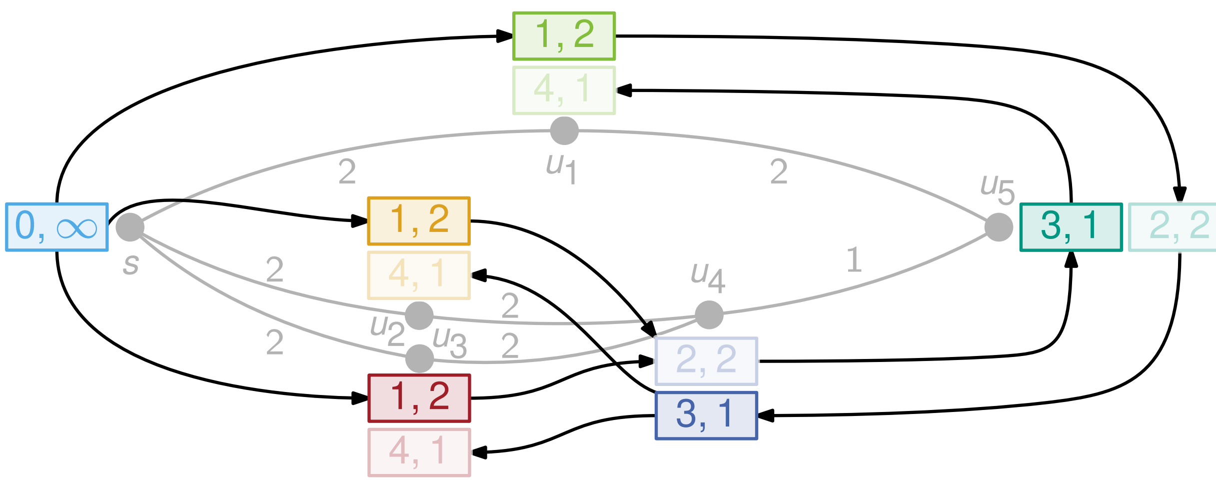


isReachable [S.354; Grastien et al., 2018]



Institut für Theoretische Informatik
Lehrstuhl Algorithmik

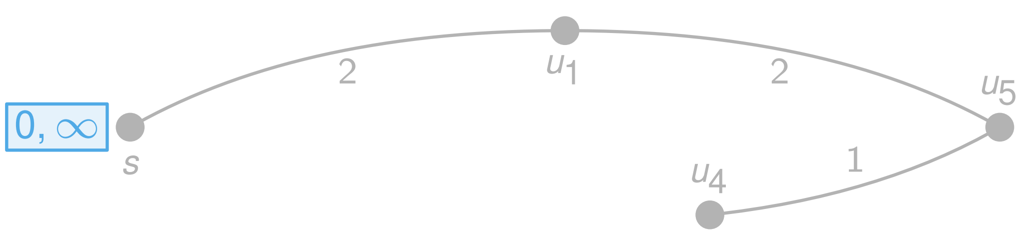
isReachable [S.354; Grastien et al., 2018]



Institut für Theoretische Informatik
Lehrstuhl Algorithmik

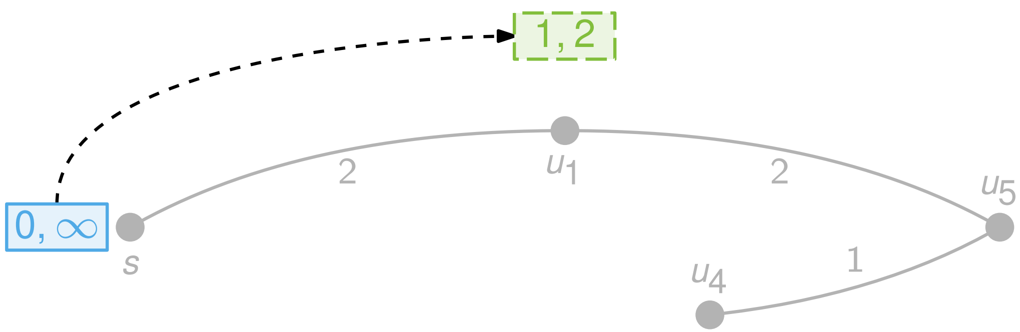
isReachable [S.354; Grastien et al., 2018]

- Funktioniert es immer?



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

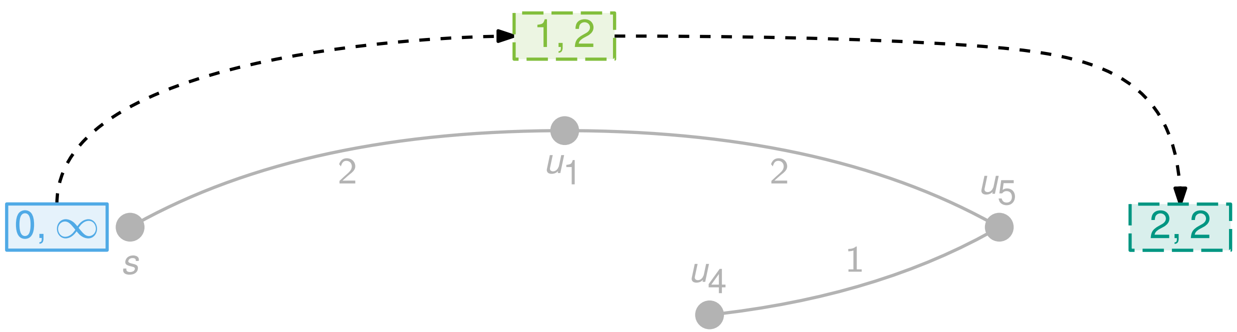
- Funktioniert es immer?



Institut für Theoretische Informatik
Lehrstuhl Algorithmik

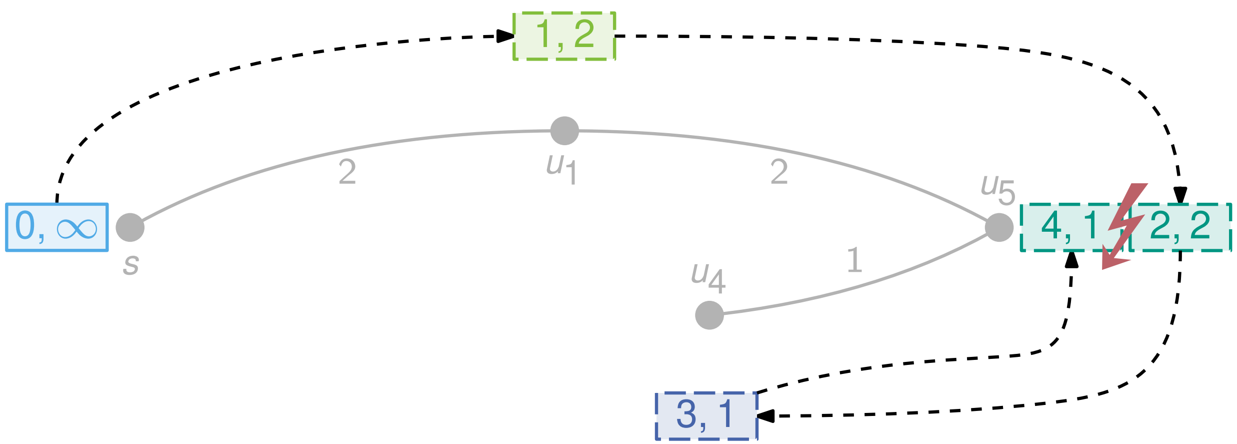
`isReachable` [S.354; Grastien et al., 2018]

- Funktioniert es immer?



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

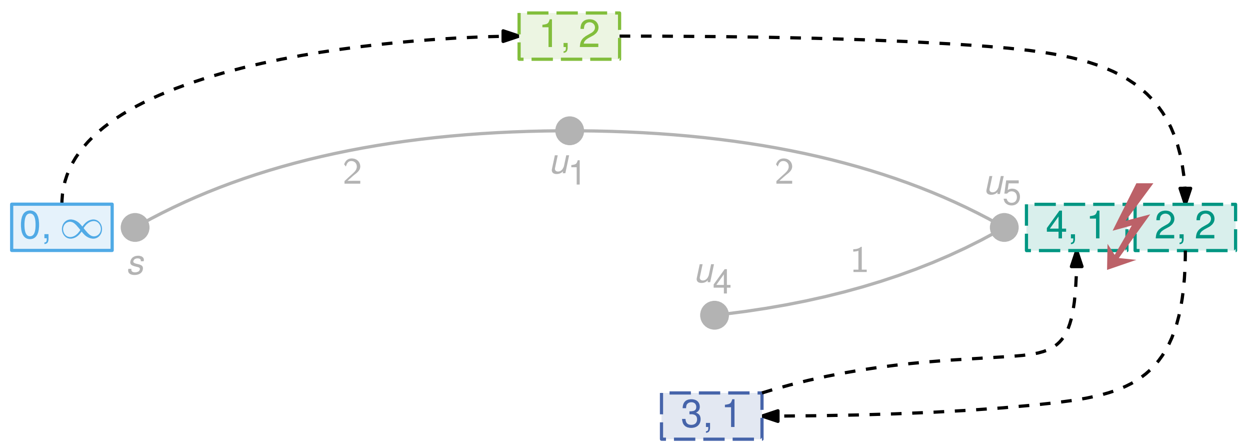
- Funktioniert es immer?



Institut für Theoretische Informatik
Lehrstuhl Algorithmik

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

- Funktioniert es immer?



$\Rightarrow$ Überprüfe, ob das Label einem einfachen Pfad entspricht

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Regenbogenpfade – FPT Algorithmen

[Satz 11; Uchizawa et al., 2013]

- FPT Algorithmus bzgl. der Farbenanzahl

Rainbow s-t-Path ($s-t-$RP)

Instanz: Ein gerichteter azyklischer Graph $G = (V, E)$, eine Färbung $c\colon V \to \mathbb{N}$, und $s, t \in V$.

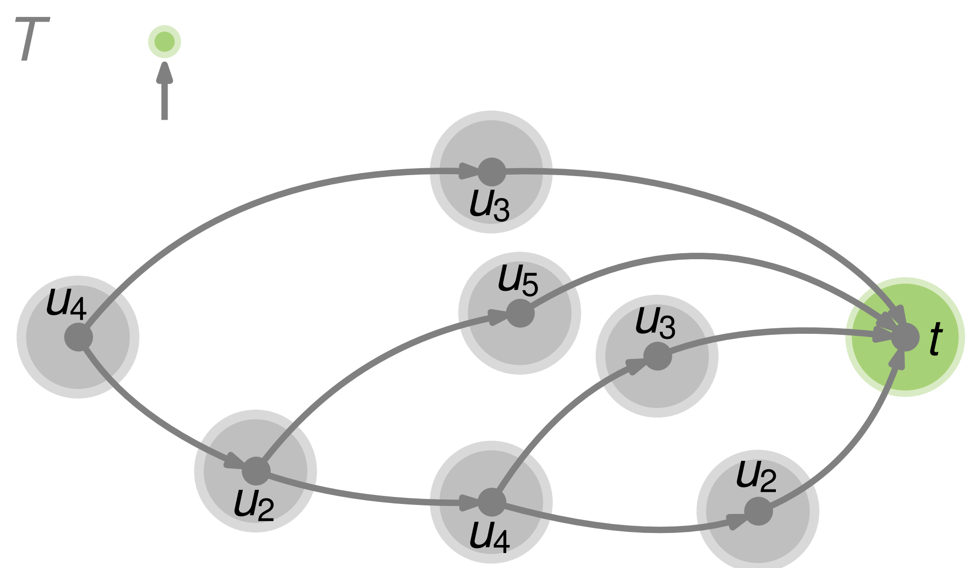
Frage: Existiert ein s-t-Pfad π in G, sodass alle Knoten von π verschiedener Farben besitzen?

Aber: n Farben $\Rightarrow$ kein Polynomialzeitalgorithmus(?)

Regenbogenpfade – FPT Algorithmen

[Satz 11; Uchizawa et al., 2013]

- FPT Algorithmus bzgl. der Farbenanzahl



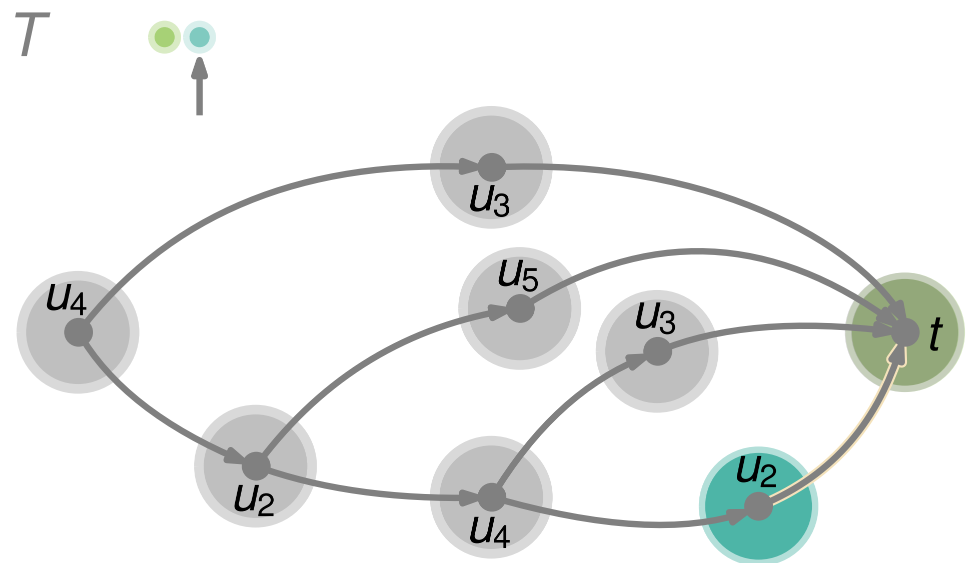
Aber: n Farben $\Rightarrow$ kein Polynomialzeitalgorithmus(?)

Regenbogenpfade – FPT Algorithmen

[Satz 11; Uchizawa et al., 2013]

Karlsruher Institut für Technologie

- FPT Algorithmus bzgl. der Farbenanzahl

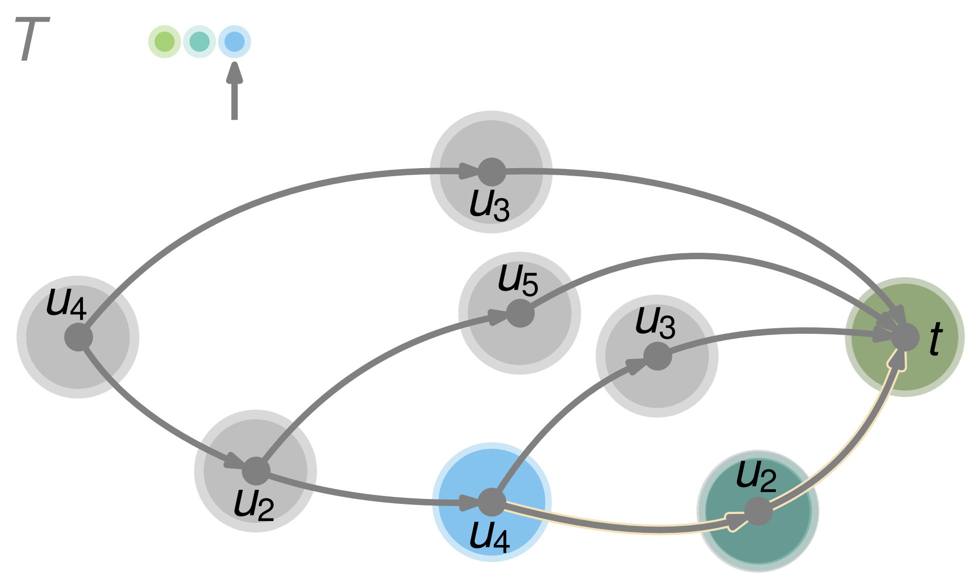


Aber: n Farben $\Rightarrow$ kein Polynomialzeitalgorithmus(?)

Regenbogenpfade – FPT Algorithmen

[Satz 11; Uchizawa et al., 2013]

- FPT Algorithmus bzgl. der Farbenanzahl

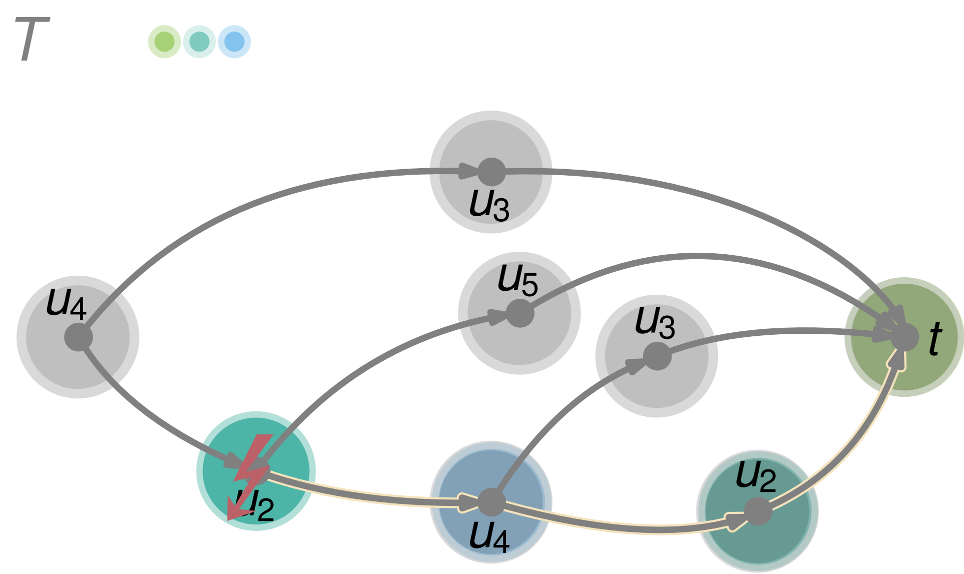


Aber: n Farben $\Rightarrow$ kein Polynomialzeitalgorithmus(?)

Regenbogenpfade – FPT Algorithmen

[Satz 11; Uchizawa et al., 2013]

- FPT Algorithmus bzgl. der Farbenanzahl



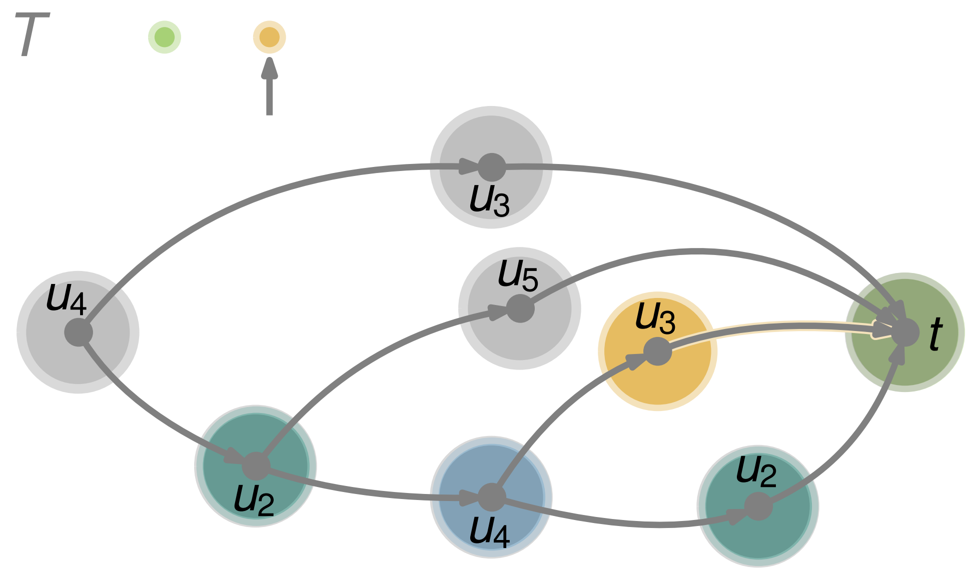
Aber: n Farben $\Rightarrow$ kein Polynomialzeitalgorithmus(?)

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Regenbogenpfade – FPT Algorithmen

[Satz 11; Uchizawa et al., 2013]

- FPT Algorithmus bzgl. der Farbenanzahl

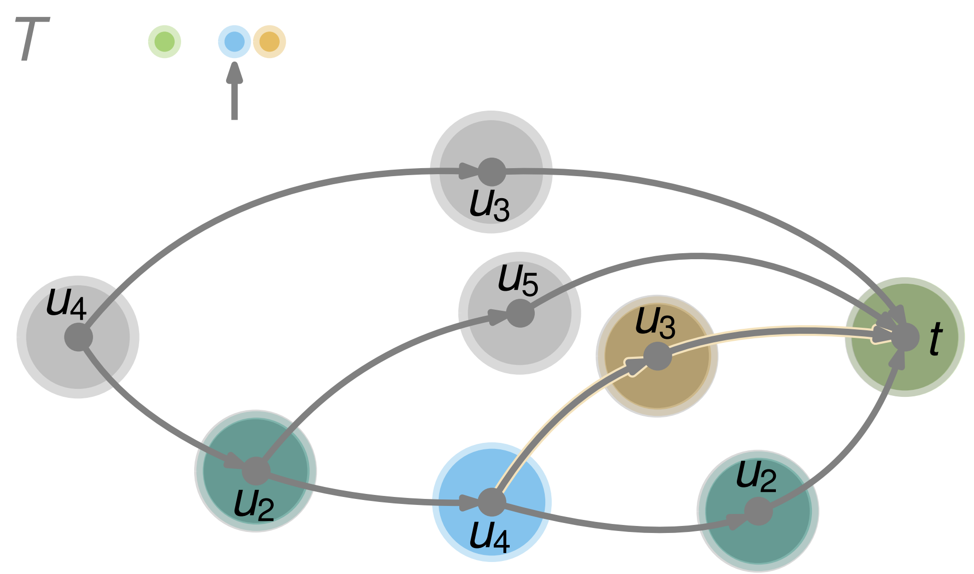


Aber: n Farben $\Rightarrow$ kein Polynomialzeitalgorithmus(?)

Regenbogenpfade – FPT Algorithmen

[Satz 11; Uchizawa et al., 2013]

- FPT Algorithmus bzgl. der Farbenanzahl

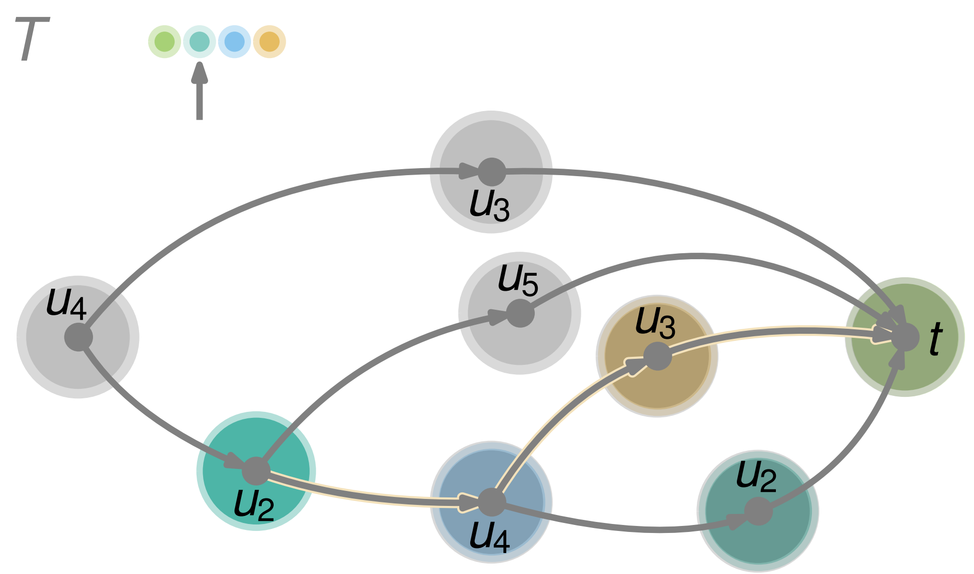


Aber: n Farben $\Rightarrow$ kein Polynomialzeitalgorithmus(?)

Regenbogenpfade – FPT Algorithmen

[Satz 11; Uchizawa et al., 2013]

- FPT Algorithmus bzgl. der Farbenanzahl

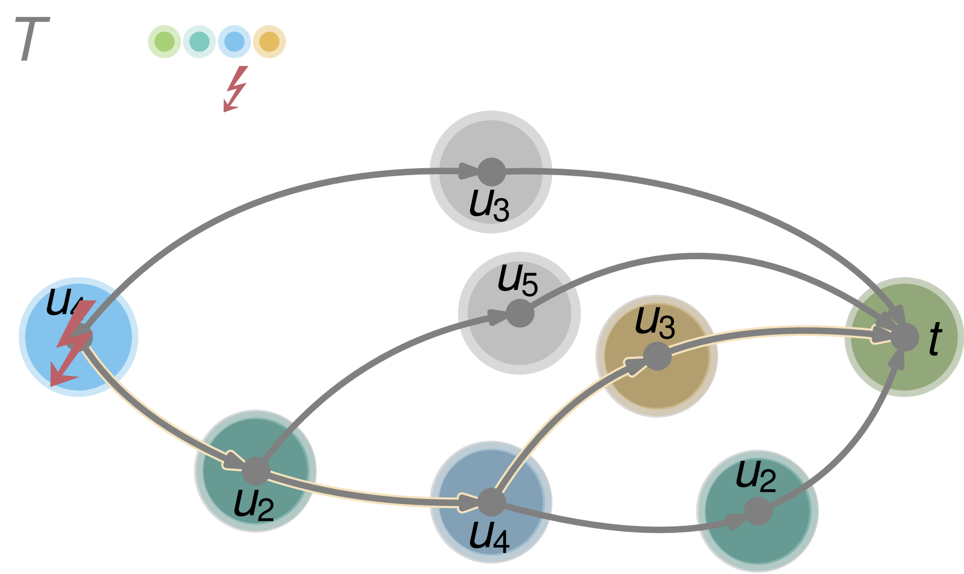


Aber: n Farben $\Rightarrow$ kein Polynomialzeitalgorithmus(?)

Regenbogenpfade – FPT Algorithmen

[Satz 11; Uchizawa et al., 2013]

- FPT Algorithmus bzgl. der Farbenanzahl

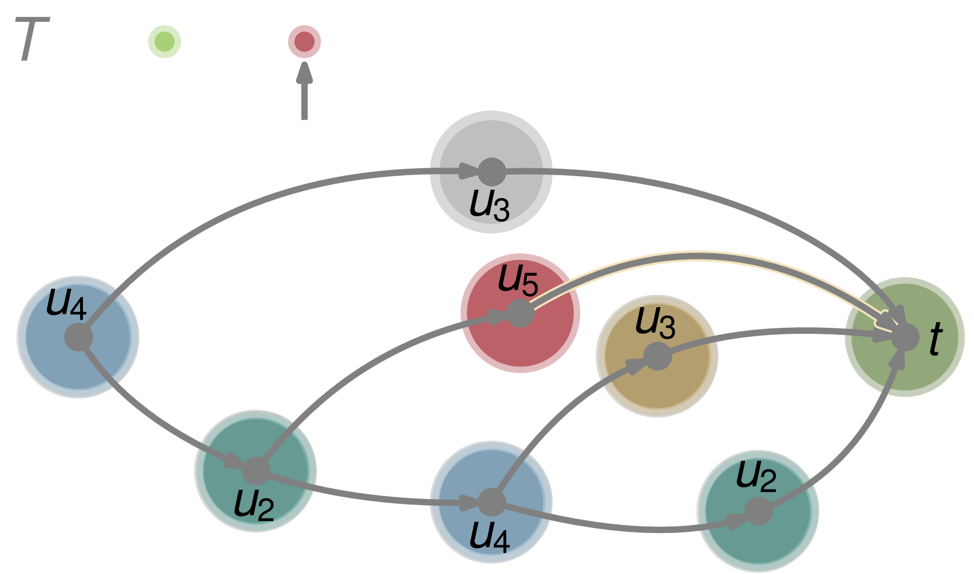


Aber: n Farben $\Rightarrow$ kein Polynomialzeitalgorithmus(?)

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

[Satz 11; Uchizawa et al., 2013]

- FPT Algorithmus bzgl. der Farbenanzahl

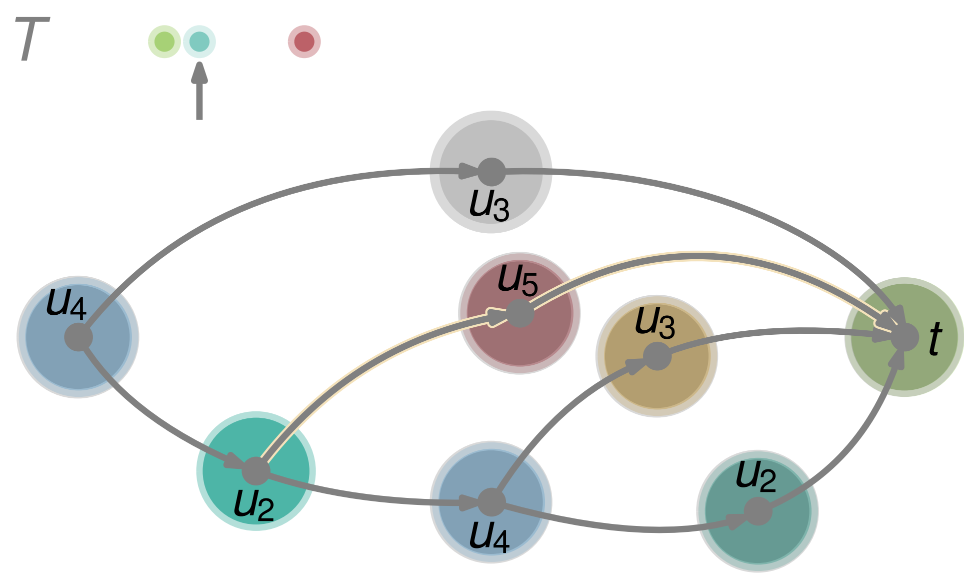


Aber: n Farben $\Rightarrow$ kein Polynomialzeitalgorithmus(?)

Regenbogenpfade – FPT Algorithmen

[Satz 11; Uchizawa et al., 2013]

- FPT Algorithmus bzgl. der Farbenanzahl

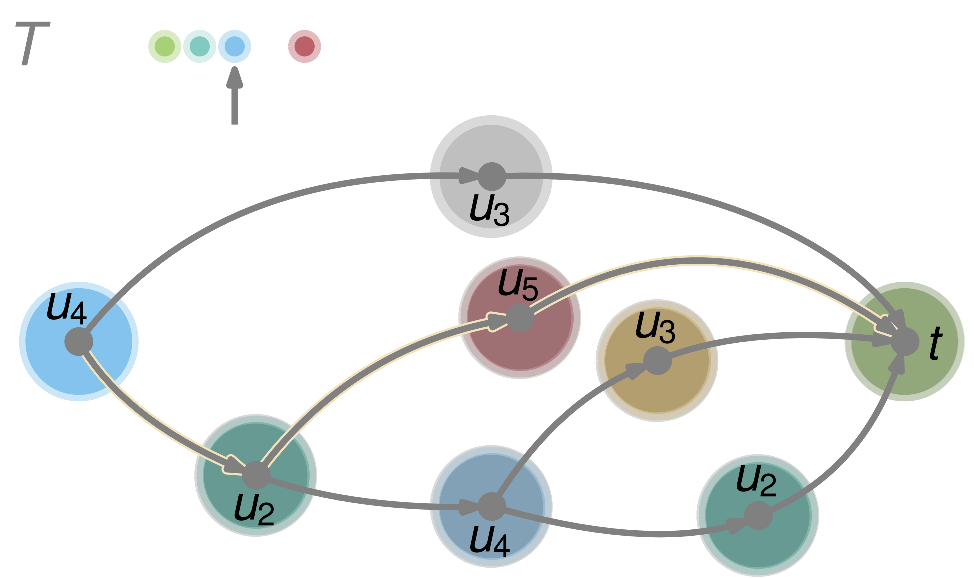


Aber: n Farben $\Rightarrow$ kein Polynomialzeitalgorithmus(?)

Regenbogenpfade – FPT Algorithmen

[Satz 11; Uchizawa et al., 2013]

- FPT Algorithmus bzgl. der Farbenanzahl

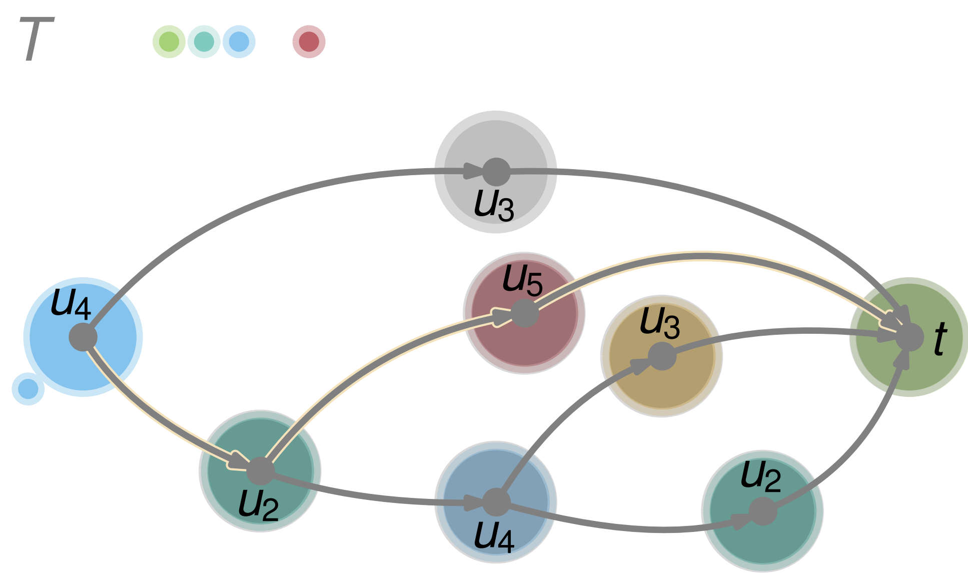


Aber: n Farben $\Rightarrow$ kein Polynomialzeitalgorithmus(?)

Regenbogenpfade – FPT Algorithmen

[Satz 11; Uchizawa et al., 2013]

- FPT Algorithmus bzgl. der Farbenanzahl



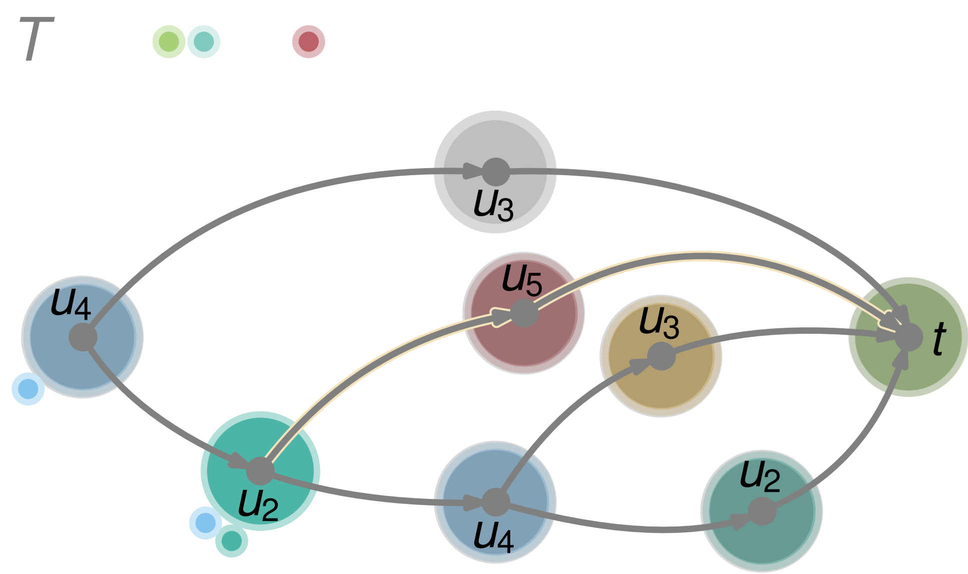
Aber: n Farben $\Rightarrow$ kein Polynomialzeitalgorithmus(?)

Regenbogenpfade – FPT Algorithmen

[Satz 11; Uchizawa et al., 2013]

Karlsruher Institut für Technologie

- FPT Algorithmus bzgl. der Farbenanzahl

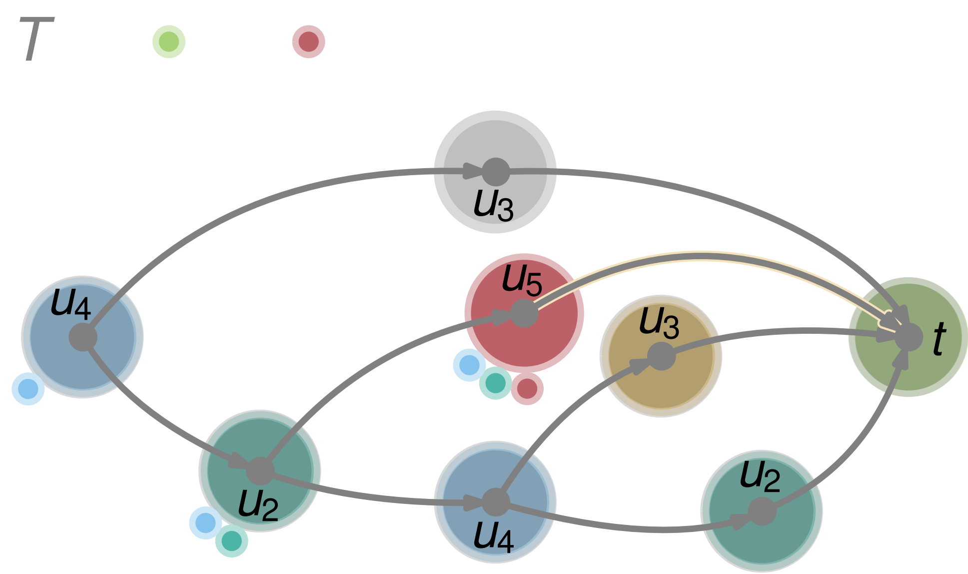


Aber: n Farben $\Rightarrow$ kein Polynomialzeitalgorithmus(?)

Regenbogenpfade – FPT Algorithmen

[Satz 11; Uchizawa et al., 2013]

- FPT Algorithmus bzgl. der Farbenanzahl



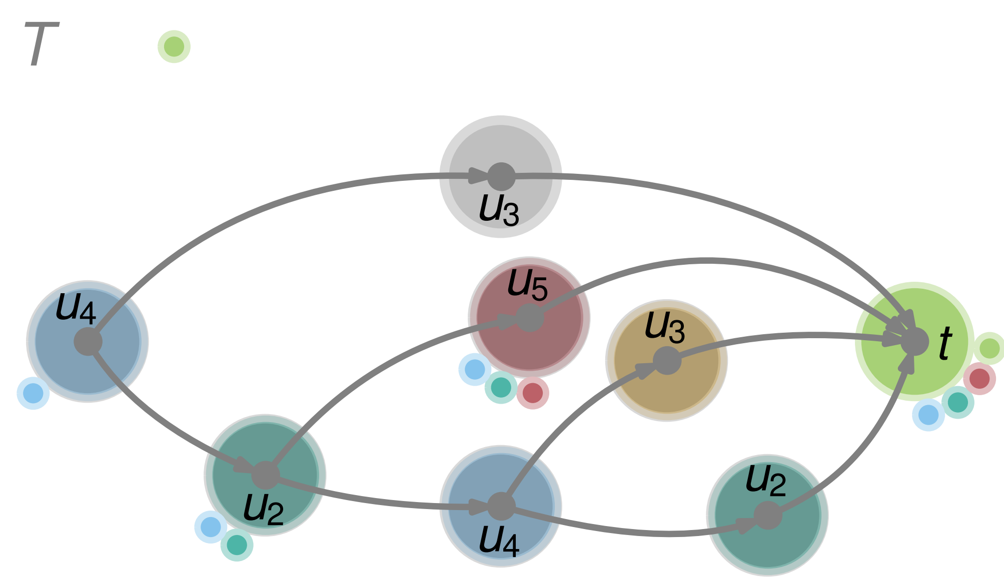
Aber: n Farben $\Rightarrow$ kein Polynomialzeitalgorithmus(?)

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Regenbogenpfade – FPT Algorithmen

[Satz 11; Uchizawa et al., 2013]

- FPT Algorithmus bzgl. der Farbenanzahl



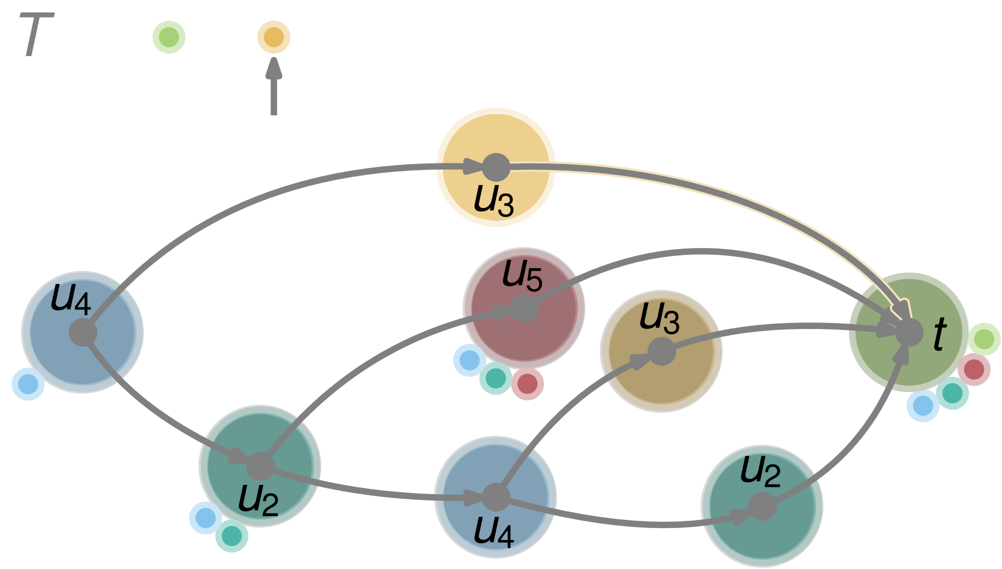
Aber: n Farben $\Rightarrow$ kein Polynomialzeitalgorithmus(?)

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Regenbogenpfade – FPT Algorithmen

[Satz 11; Uchizawa et al., 2013]

- FPT Algorithmus bzgl. der Farbenanzahl



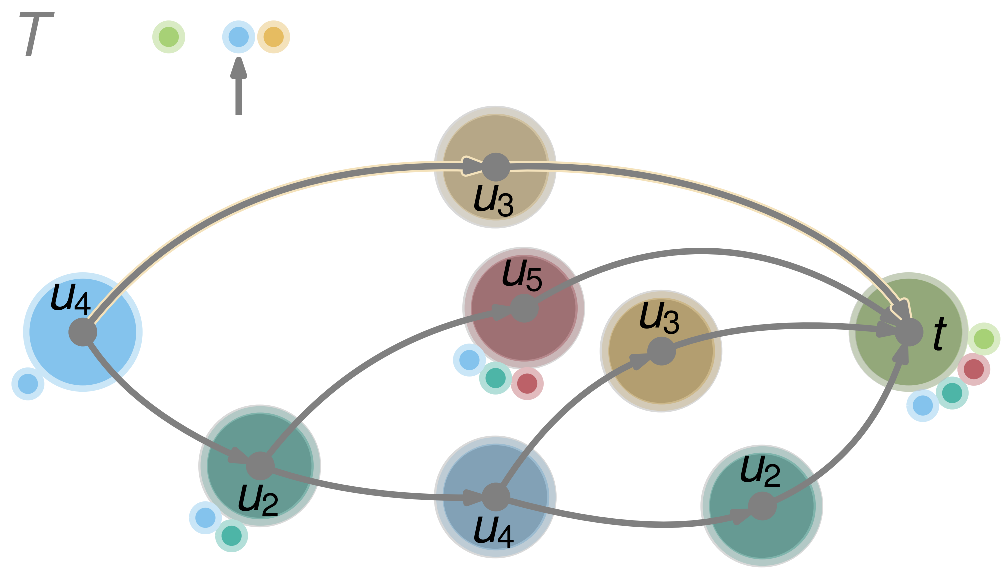
Aber: n Farben $\Rightarrow$ kein Polynomialzeitalgorithmus(?)

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Regenbogenpfade – FPT Algorithmen

[Satz 11; Uchizawa et al., 2013]

- FPT Algorithmus bzgl. der Farbenanzahl



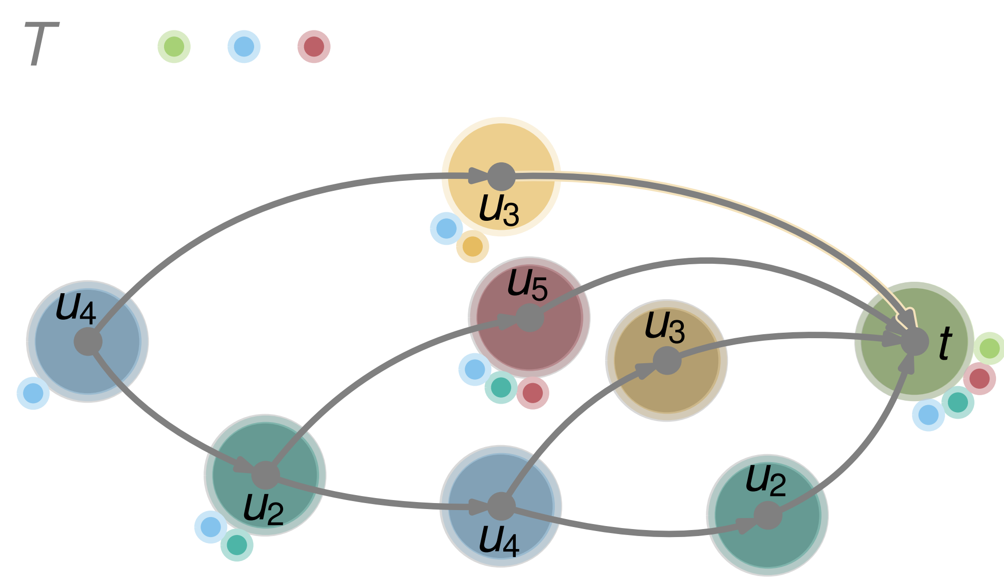
Aber: n Farben $\Rightarrow$ kein Polynomialzeitalgorithmus(?)

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Regenbogenpfade – FPT Algorithmen

[Satz 11; Uchizawa et al., 2013]

- FPT Algorithmus bzgl. der Farbenanzahl

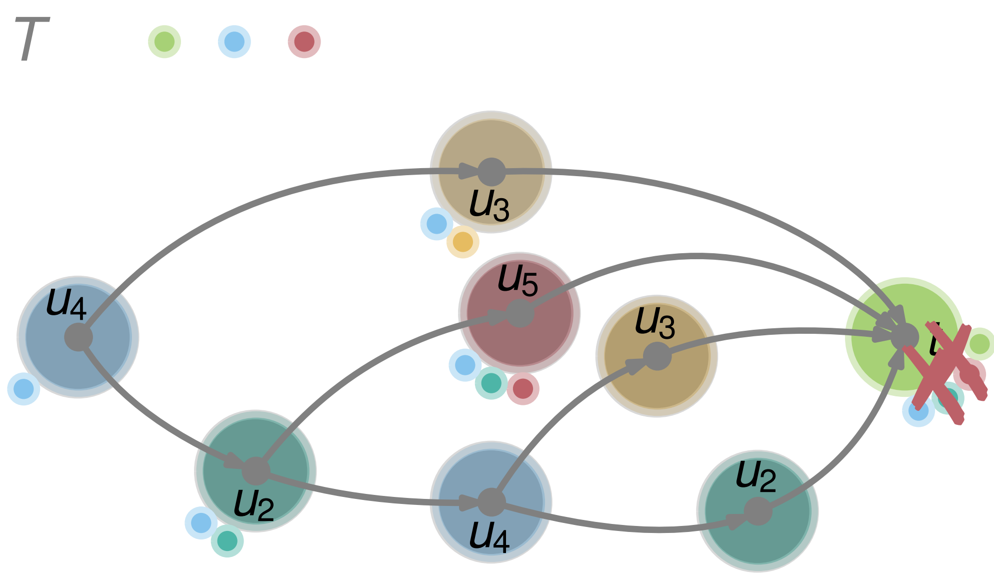


Aber: n Farben $\Rightarrow$ kein Polynomialzeitalgorithmus(?)

Regenbogenpfade – FPT Algorithmen

[Satz 11; Uchizawa et al., 2013]

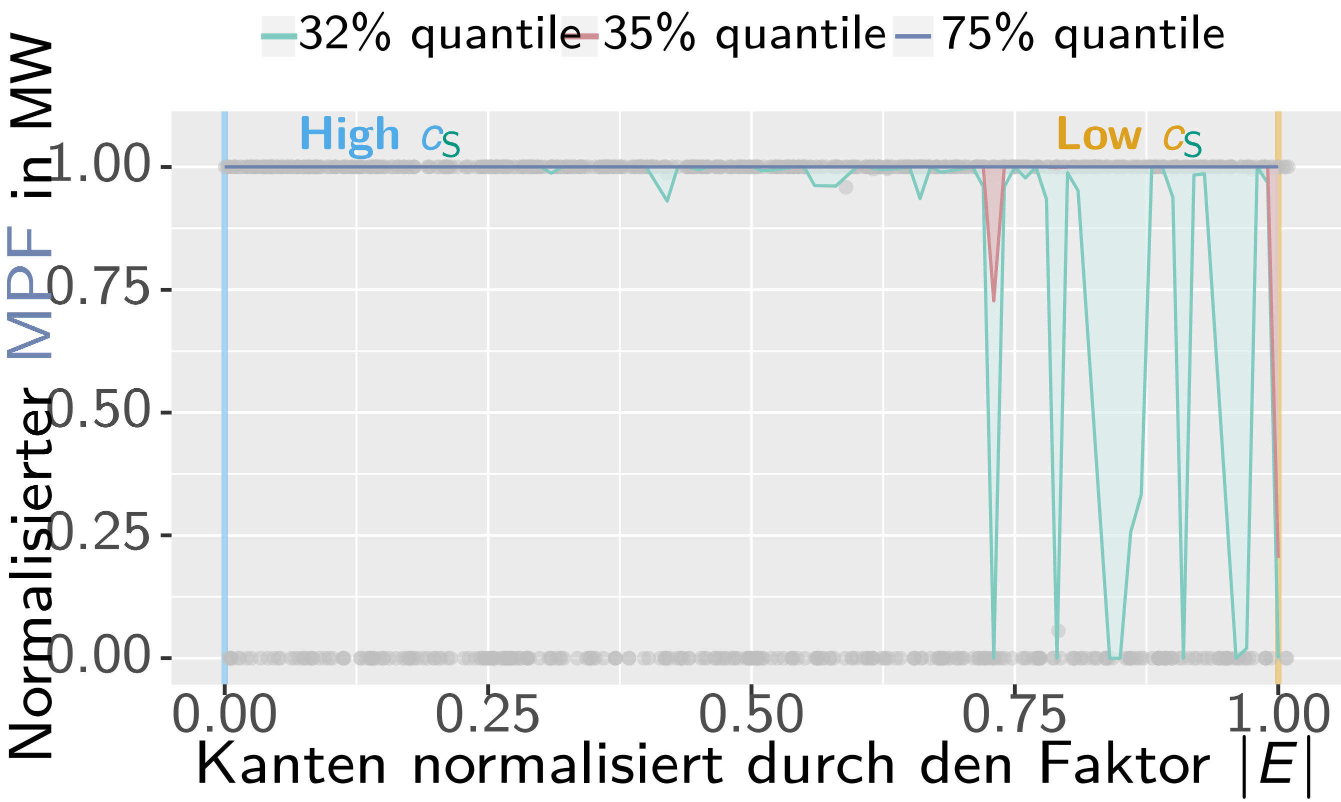
- FPT Algorithmus bzgl. der Farbenanzahl



Aber: n Farben $\Rightarrow$ kein Polynomialzeitalgorithmus(?)

Simulationen [Seite 349; Grastien et al., 2018]

- Simulationen auf dem NESTA Benchmarksets, die realistischer als die IEEE Benchmarksets sind, z.B., mit Hinblick auf thermische Leitungs-begrenzungen



Der MPF sinkt hauptsächlich für Kanten mit kleiner Zentralität c_S.

Simulationen [Seite 349; Grastien et al., 2018]

- Simulationen auf dem NESTA Benchmarksets, die realistischer als die IEEE Benchmarksets sind, z.B., mit Hinblick auf thermische Leitungsbegrenzungen

Auf **allgemeinen Netzwerken** ist die *Schaltzentralität* $c_S : E \to \mathbb{R}_{\geq 0}$ definiert durch

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

wobei $\sigma_{\mathrm{DTP}}(s,t,e)$ die Anzahl von DTP-Pfaden zwischen s und t ist, die die Kante e nutzen, $\sigma_{\mathrm{DTP}}(s,t)$ ist die Gesamtanzahl von DTP-Pfaden von s nach t, und $m_B = |V|(|V|-1)$ wird zur Normalisierung verwendet.

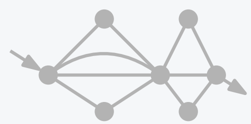
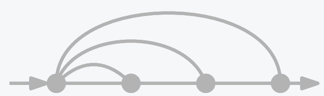
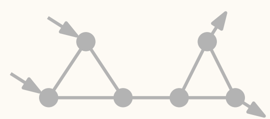
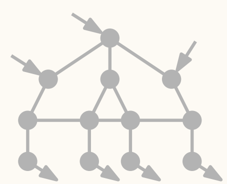
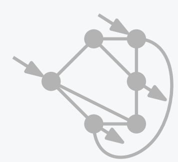
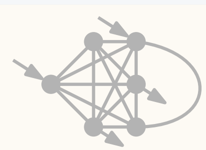
Kanten normalisiert durch den Faktor $|E|$

Der MPF sinkt hauptsächlich für Kanten mit kleiner Zentralität c_S.

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Überblick über die MTSF Ergebnisse

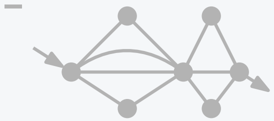
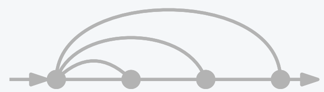
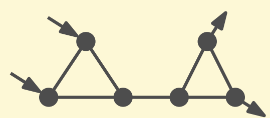
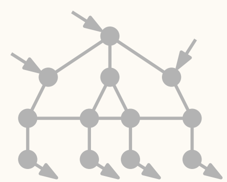
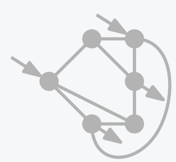
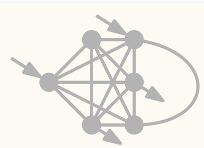
Karlsruher Institut für Technologie

	Graphstruktur		Komplexität	Algorithmen
ein Erzeuger, ein Verbraucher	Penrose-Minoren-freie Graphen		Polynomialzeit lösbar	DTP ✓
	Serienparallele Graphen		NP-schwer	✗
beliebige Erzeuger, beliebige Verbraucher	Kakteen mit Maximalgrad 3		NP-schwer [Lehmann et al., 2014]	2-approx. ✓
	2-Level-Bäume		NP-schwer [Lehmann et al., 2014]	✗
	Planare Graphen mit Maximalgrad 3		stark NP-schwer [Lehmann et al., 2014]	✗
$\lvert V_G \rvert = 2$, $\lvert V_D \rvert = 2$	Beliebige Graphen		nicht-APX [Lehmann et al., 2014]	✗

Komplexität

Überblick über die MTSF Ergebnisse

Karlsruher Institut für Technologie

	Graphstruktur		Komplexität	Algorithmen
ein Erzeuger, ein Verbraucher	Penrose-Minoren-freie Graphen		Polynomialzeit lösbar	DTP ✓
	Serienparallele Graphen		NP-schwer	✗
beliebige Erzeuger, beliebige Verbraucher	Kakteen mit Maximalgrad 3		NP-schwer [Lehmann et al., 2014]	2-approx. ✓
	2-Level-Bäume		NP-schwer [Lehmann et al., 2014]	✗
	Planare Graphen mit Maximalgrad 3		stark NP-schwer [Lehmann et al., 2014]	✗
$\lvert V_G\rvert=2$, $\lvert V_D\rvert=2$	Beliebige Graphen		nicht-APX [Lehmann et al., 2014]	✗

Komplexität

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

NP-Vollständigkeit für Entscheidungsprobleme

[S.8; TGI, VL 19.11.2019]

Lemma 3 [S.8; TGI, VL 19.11.2019]

Entscheidungsproblem $\mathcal{P}_1$ ist polynomial transformierbar in das Entscheidungsproblem $\mathcal{P}_2$, wenn eine Funktion $f\colon D_{\mathcal{P}_1} \to D_{\mathcal{P}_2}$ existiert mit folgenden Eigenschaften

- f ist durch einen polynomialen Algorithmus berechenbar
- für alle $I \in D_{\mathcal{P}_1} : I \in J_{\mathcal{P}_1} \Leftrightarrow f(I) \in J_{\mathcal{P}_2}$

Wir schreiben dann $\mathcal{P}_1 \propto \mathcal{P}_2$.

Lemma 4 [S.8; TGI, VL 19.11.2019]

Ein Entscheidungsproblem $\mathcal{P}$ heißt NP-vollständig, falls gilt

- $\mathcal{P} \in \text{NP}$ und
- für alle $\mathcal{P}' \in \text{NP}$ gilt $\mathcal{P}' \propto \mathcal{P}$.

Subset Sum Problem (SSP)

Entscheidungsproblem Subset Sum (SSP)

Instanz: Eine endliche Menge von Zahlen $W = \{w_1, w_2, \ldots, w_n\}$ mit $w_i \in \mathbb{N}$ und $k \in \mathbb{N}$.

Frage: Existiert eine Menge von Elementen $x_1, x_2, \ldots, x_n \in \{0, 1\}$, sodass $\sum_{j=1}^{n} w_j x_j = k$?

Subset Sum Problem (SSP)

Entscheidungsproblem Subset Sum (SSP)

Instanz: Eine endliche Menge von Zahlen $W = \{w_1, w_2, \dots, w_n\}$ mit $w_i \in \mathbb{N}$ und $k \in \mathbb{N}$.

Frage: Existiert eine Menge von Elementen $x_1, x_2, \dots, x_n \in \{0, 1\}$, sodass $\sum_{j=1}^{n} w_j x_j = k$?

Beispielinstanz

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

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Subset Sum Problem (SSP)

Karlsruher Institut für Technologie

Entscheidungsproblem Subset Sum (SSP)

Instanz: Eine endliche Menge von Zahlen $W = \{w_1, w_2, \ldots, w_n\}$ mit $w_i \in \mathbb{N}$ und $k \in \mathbb{N}$.

Frage: Existiert eine Menge von Elementen $x_1, x_2, \ldots, x_n \in \{0, 1\}$, sodass $\sum_{j=1}^{n} w_j x_j = k$?

Beispielinstanz

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

Lösung

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

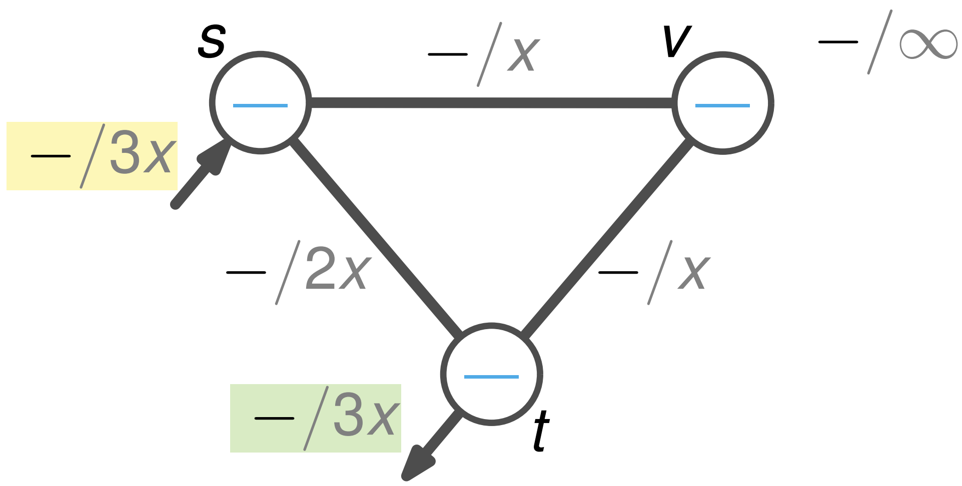
Institut für Theoretische Informatik
Lehrstuhl Algorithmik

MTSF auf Kakteen ist NP-schwer

[Lehmann et al., 2014]

- MTSF für Kaktusnetzwerke mit einem Maximalgrad von 3 ist NP-schwer
- Reduktion von Subset Sum
- Schaltungswahlnetzwerk (SCN) ist ein Gadget, welches Entscheidungen kodiert, die ein Netzwerk repräsentieren $\mathrm{SCN}_{\ell,v} = \left(\{s, t, v\}, E, \underline{p_d} \coloneqq 3\ell, \overline{p_d} \coloneqq 3\ell, \overline{x} \coloneqq 3\ell, \mathrm{cap}\right)$

Schaltungswahlnetzwerk SCN

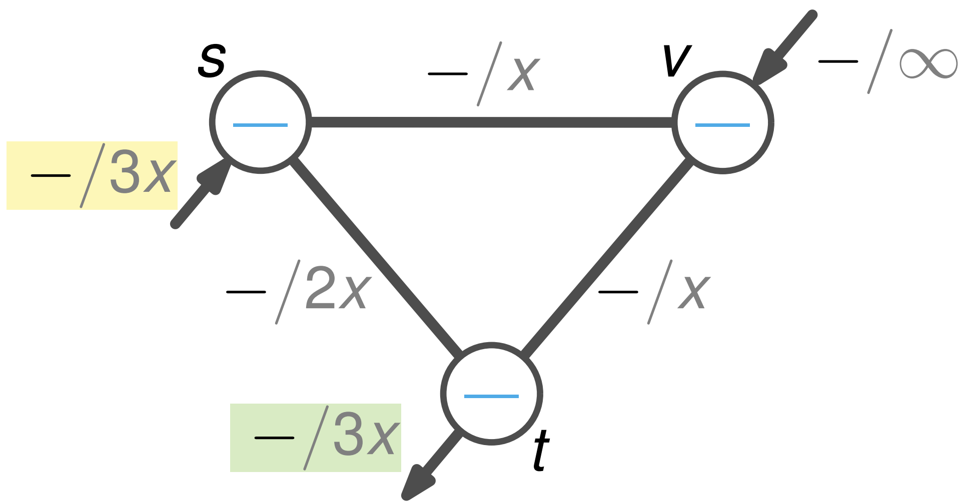


MTSF auf Kakteen ist NP-schwer

[Lehmann et al., 2014]

- MTSF für Kaktusnetzwerke mit einem Maximalgrad von 3 ist NP-schwer
- Reduktion von Subset Sum
- Schaltungswahlnetzwerk (SCN) ist ein Gadget, welches Entscheidungen kodiert, die ein Netzwerk repräsentieren $\mathrm{SCN}_{\ell,v} = \left(\{s, t, v\}, E, \underline{p_d} \coloneqq 3\ell, \overline{p_d} \coloneqq 3\ell, \overline{x} \coloneqq 3\ell, \mathrm{cap}\right)$

Schaltungswahlnetzwerk $\mathrm{SCN}^{-}_{x,v}$

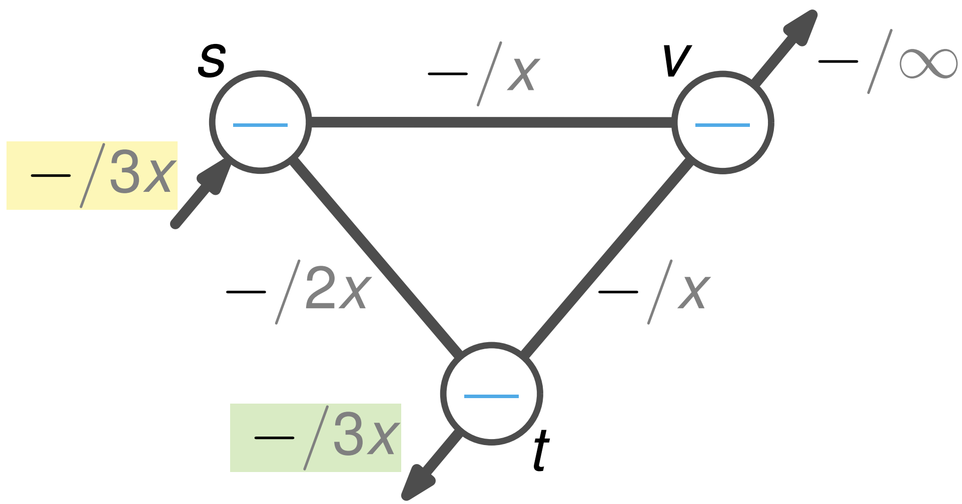


MTSF auf Kakteen ist NP-schwer

[Lehmann et al., 2014]

- MTSF für Kaktusnetzwerke mit einem Maximalgrad von 3 ist NP-schwer
- Reduktion von Subset Sum
- Schaltungswahlnetzwerk (SCN) ist ein Gadget, welches Entscheidungen kodiert, die ein Netzwerk repräsentieren $\mathrm{SCN}_{\ell,v} = \left(\{s,t,v\}, E, \underline{p_d} \coloneqq 3\ell, \overline{p_d} \coloneqq 3\ell, \overline{x} \coloneqq 3\ell, \mathrm{cap}\right)$

Schaltungswahlnetzwerk $\mathrm{SCN}^{+}_{x,v}$

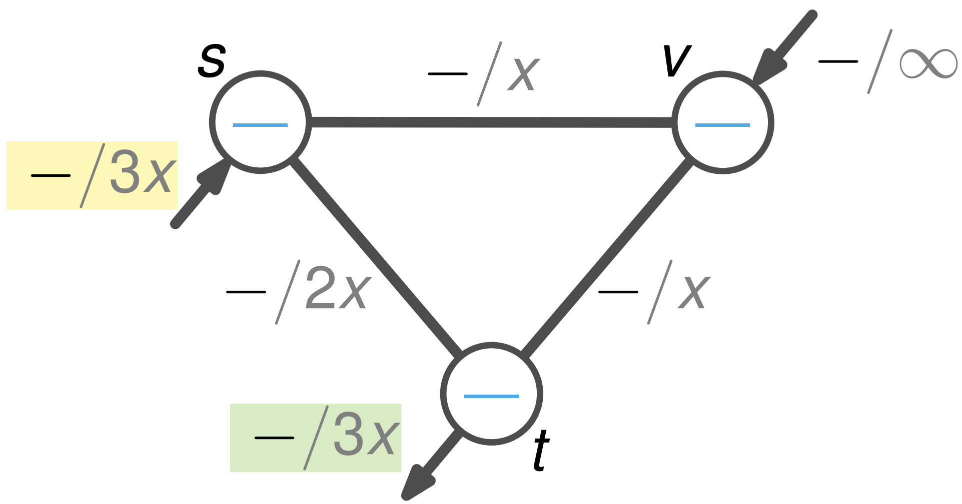


MTSF auf Kakteen ist NP-schwer

[Lehmann et al., 2014]

- MTSF für Kaktusnetzwerke mit einem Maximalgrad von 3 ist NP-schwer
- Reduktion von Subset Sum
- Schaltungswahlnetzwerk (SCN) ist ein Gadget, welches Entscheidungen kodiert, die ein Netzwerk repräsentieren $\mathrm{SCN}_{\ell,v} = \left(\{s,t,v\}, E, \underline{p_d} := 3\ell, \overline{p_d} := 3\ell, \overline{x} := 3\ell, \mathrm{cap}\right)$

Schaltungswahlnetzwerk $\mathrm{SCN}^{-}_{x,v}$



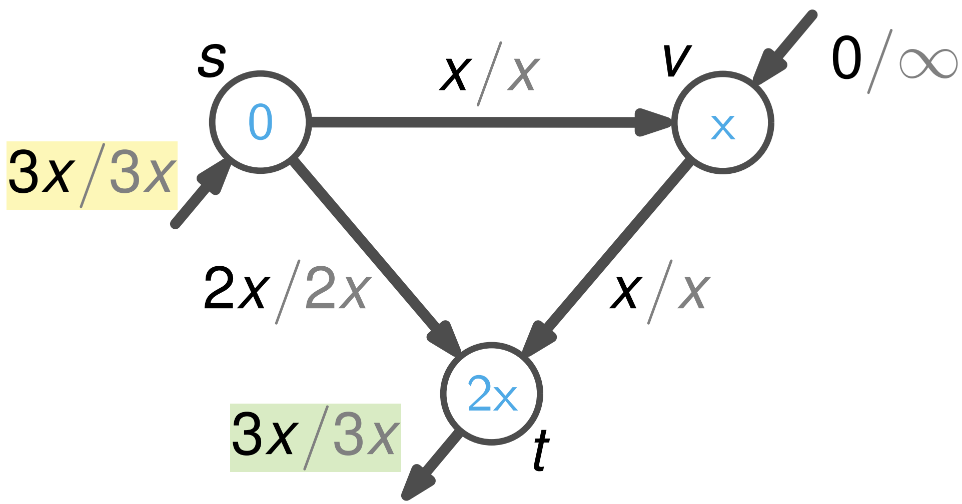
Institut für Theoretische Informatik
Lehrstuhl Algorithmik

MTSF auf Kakteen ist NP-schwer

[Lehmann et al., 2014]

- MTSF für Kaktusnetzwerke mit einem Maximalgrad von 3 ist NP-schwer
- Reduktion von Subset Sum
- Schaltungswahlnetzwerk (SCN) ist ein Gadget, welches Entscheidungen kodiert, die ein Netzwerk repräsentieren $\mathrm{SCN}_{\ell,v} = \left(\{s,t,v\}, E, \underline{p_d} := 3\ell, \overline{p_d} := 3\ell, \overline{x} := 3\ell, \mathrm{cap}\right)$

Schaltungswahlnetzwerk $\mathrm{SCN}^{-}_{x,v}$



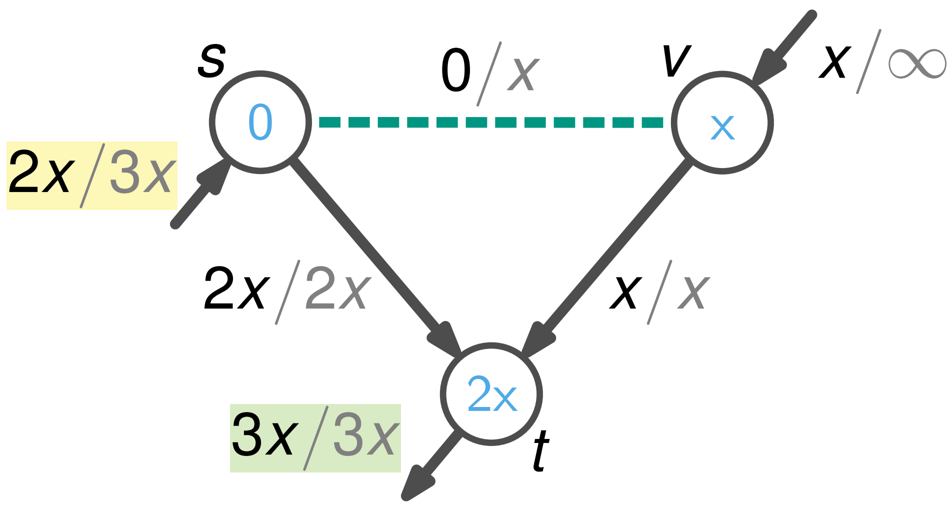
Institut für Theoretische Informatik
Lehrstuhl Algorithmik

MTSF auf Kakteen ist NP-schwer

[Lehmann et al., 2014]

- MTSF für Kaktusnetzwerke mit einem Maximalgrad von 3 ist NP-schwer
- Reduktion von Subset Sum
- Schaltungswahlnetzwerk (SCN) ist ein Gadget, welches Entscheidungen kodiert, die ein Netzwerk repräsentieren $\text{SCN}_{\ell,v} = \left(\{s,t,v\}, E, \underline{p_d} \coloneqq 3\ell, \overline{p_d} \coloneqq 3\ell, \overline{x} \coloneqq 3\ell, \text{cap}\right)$

Schaltungswahlnetzwerk $\text{SCN}^{-}_{x,v}$

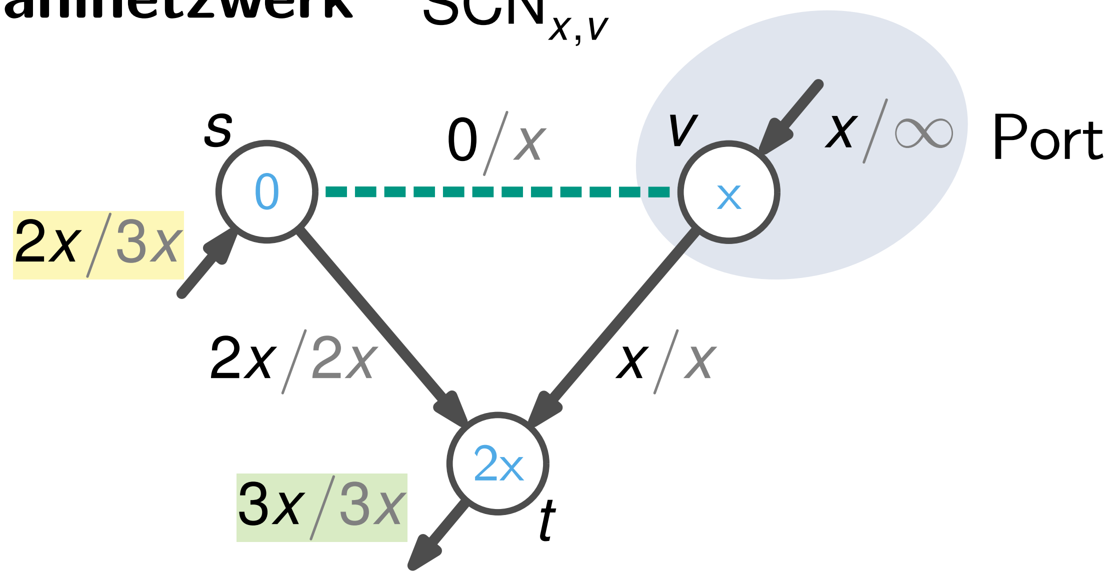


MTSF auf Kakteen ist NP-schwer

[Lehmann et al., 2014]

- MTSF für Kaktusnetzwerke mit einem Maximalgrad von 3 ist NP-schwer
- Reduktion von Subset Sum
- Schaltungswahlnetzwerk (SCN) ist ein Gadget, welches Entscheidungen kodiert, die ein Netzwerk repräsentieren $\mathrm{SCN}_{\ell,v} = \left(\{s,t,v\}, E, \underline{p_d} \coloneqq 3\ell, \overline{p_d} \coloneqq 3\ell, \overline{x} \coloneqq 3\ell, \mathrm{cap}\right)$

Schaltungswahlnetzwerk $\mathrm{SCN}^{-}_{x,v}$



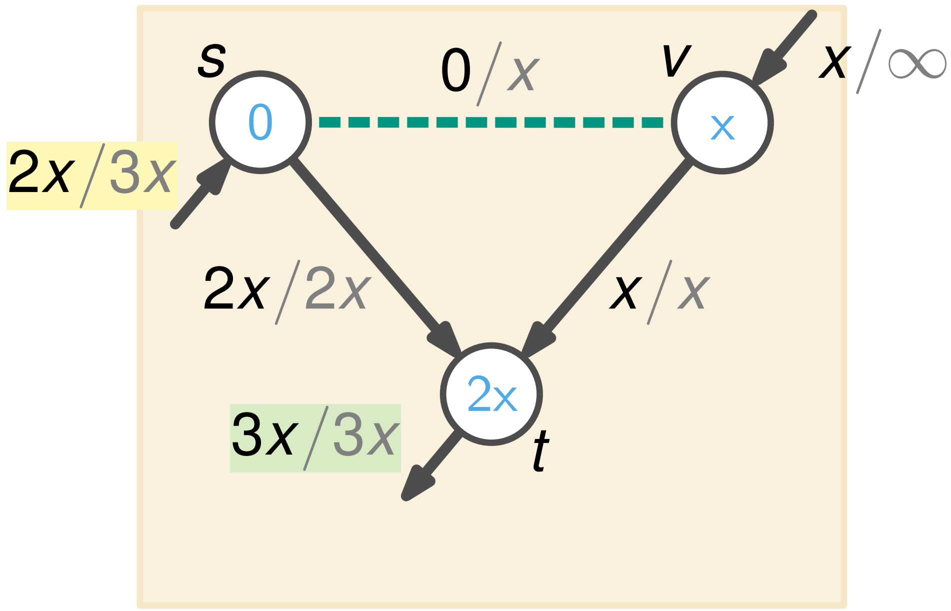
Institut für Theoretische Informatik
Lehrstuhl Algorithmik

MTSF auf Kakteen ist NP-schwer

[Lehmann et al., 2014]

- MTSF für Kaktusnetzwerke mit einem Maximalgrad von 3 ist NP-schwer
- Reduktion von Subset Sum
- Schaltungswahlnetzwerk (SCN) ist ein Gadget, welches Entscheidungen kodiert, die ein Netzwerk repräsentieren $\mathrm{SCN}_{\ell,v} = \left(\{s,t,v\}, E, \underline{p_d} := 3\ell, \overline{p_d} := 3\ell, \overline{x} := 3\ell, \mathrm{cap}\right)$

Schaltungswahlnetzwerk $\mathrm{SCN}^{-}_{x,v}$

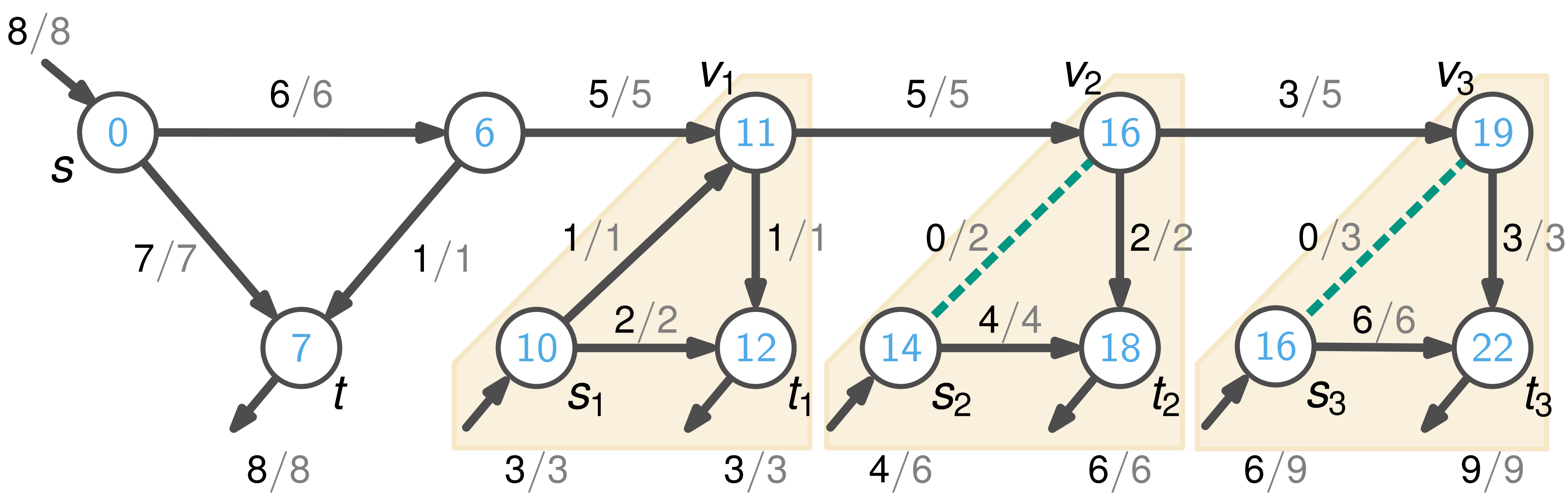


MTSF auf Kakteen ist NP-schwer

[Lehmann et al., 2014]

- MTSF für Kaktusnetzwerke mit einem Maximalgrad von 3 ist NP-schwer
- Reduktion von Subset Sum
- Schaltungswahlnetzwerk (SCN) ist ein Gadget, welches Entscheidungen kodiert, die ein Netzwerk repräsentieren $\mathrm{SCN}_{\ell,v} = \left(\{s,t,v\}, E, \underline{p_d} \coloneqq 3\ell, \overline{p_d} \coloneqq 3\ell, \overline{x} \coloneqq 3\ell, \mathrm{cap}\right)$

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



Institut für Theoretische Informatik
Lehrstuhl Algorithmik

2-Approximation auf Kakteen

Karlsruher Institut für Technologie

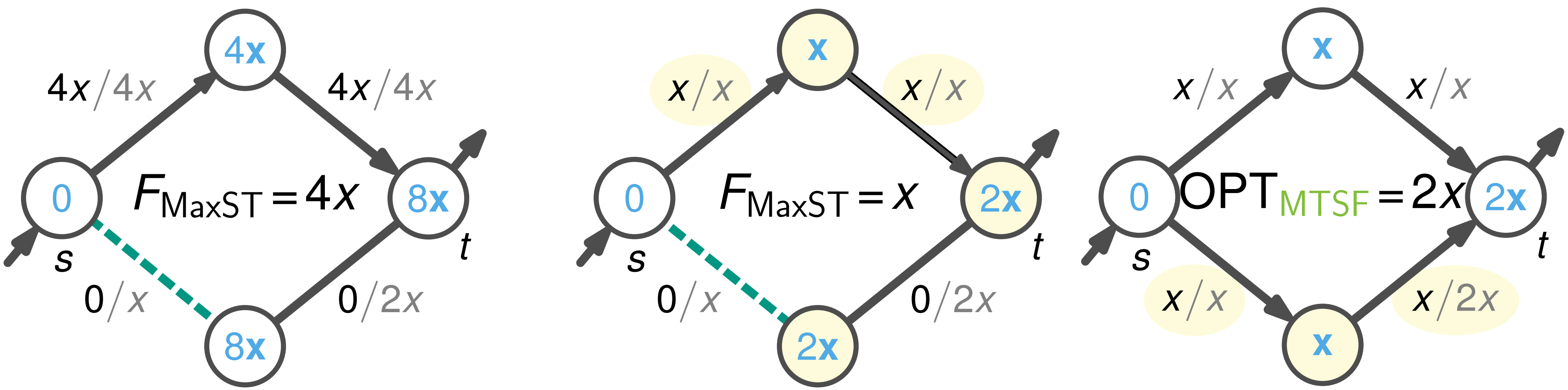
Beschreibung

- Entferne von jedem Kreis die Kante mit der kleinsten Kapazität

$\Leftrightarrow$ dem Maximum Spanning Tree (MaxST)

MaxST auf Kakteen

- MTSF ist NP-schwer auf Kakteen [Lehmann et al., 2014]



Satz 5 [Seite 348; Grastien et al., 2018]

MaxST ist ein 2-Approximationsalgorithmus für das MFP und MTSF-Problem auf Kakteen.

2-Approximation auf Kakteen

Karlsruher Institut für Technologie

Beschreibung

- Entferne von jedem Kreis die Kante mit der kleinsten Kapazität

⇔ dem Maximum Spanning Tree (MaxST)

MaxST auf Kakteen

- MTSF ist NP-schwer auf Kakteen [Lehmann et al., 2014]

Auf Kakteen läuft der MaxST-Algorithmus in $\mathcal{O}(|V|)$ Zeit.

Satz 5 [Seite 348; Grastien et al., 2018]

MaxST ist ein 2-Approximationsalgorithmus für das MFP und MTSF-Problem auf Kakteen.

2-Approximation auf Kakteen

Daten: Ein Netzwerk $\mathcal{N} = (G, V_G, V_D, \text{cap}, b)$.
Ergebnis: $\text{OPT}_{\text{MPFP}}(\mathcal{N} - S)$, und geschaltete Kanten S.

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

Lemma 6 [Seite 347; Grastien et al., 2018]

Sei $\mathcal{N} = (G, V_G, V_D, \text{cap}, b)$ ein Energienetz und sei S eine Menge $\arg\min_{\forall e \in c} \text{cap}(e)$ von geschalteten Kanten für alle Kreise $c \in C$. Dann existiert ein elektrisch zulässiger Fluss f' auf $\mathcal{N} - S$, sodass $F(f') = \frac{1}{2}\,\text{OPT}_{\text{MFP}}(\mathcal{N})$.

Beweis.

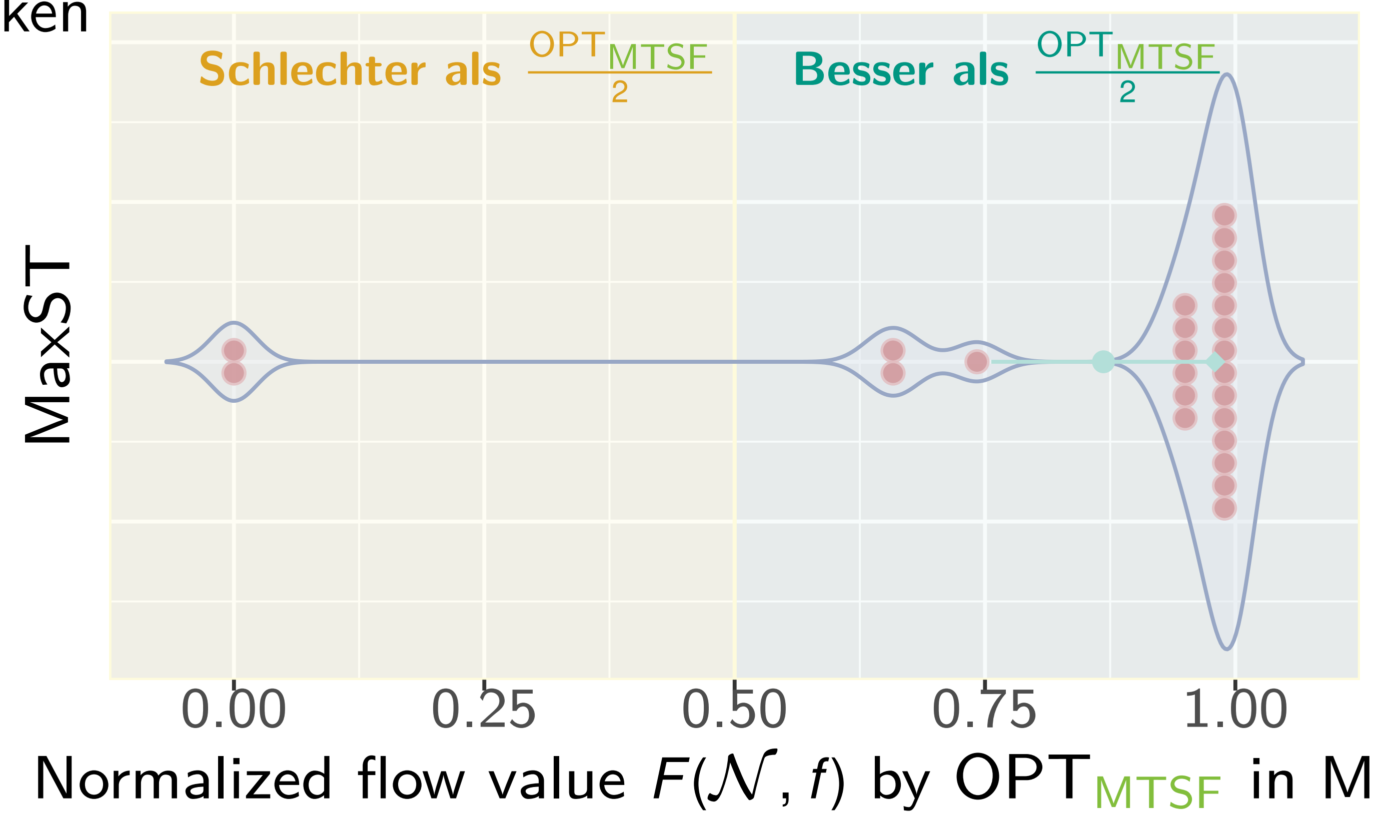
$$|f(e)| = \left|\frac{1}{2}\, f^\star(e)\right| \leq \frac{1}{2}\, \text{cap}(e),$$

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

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

□

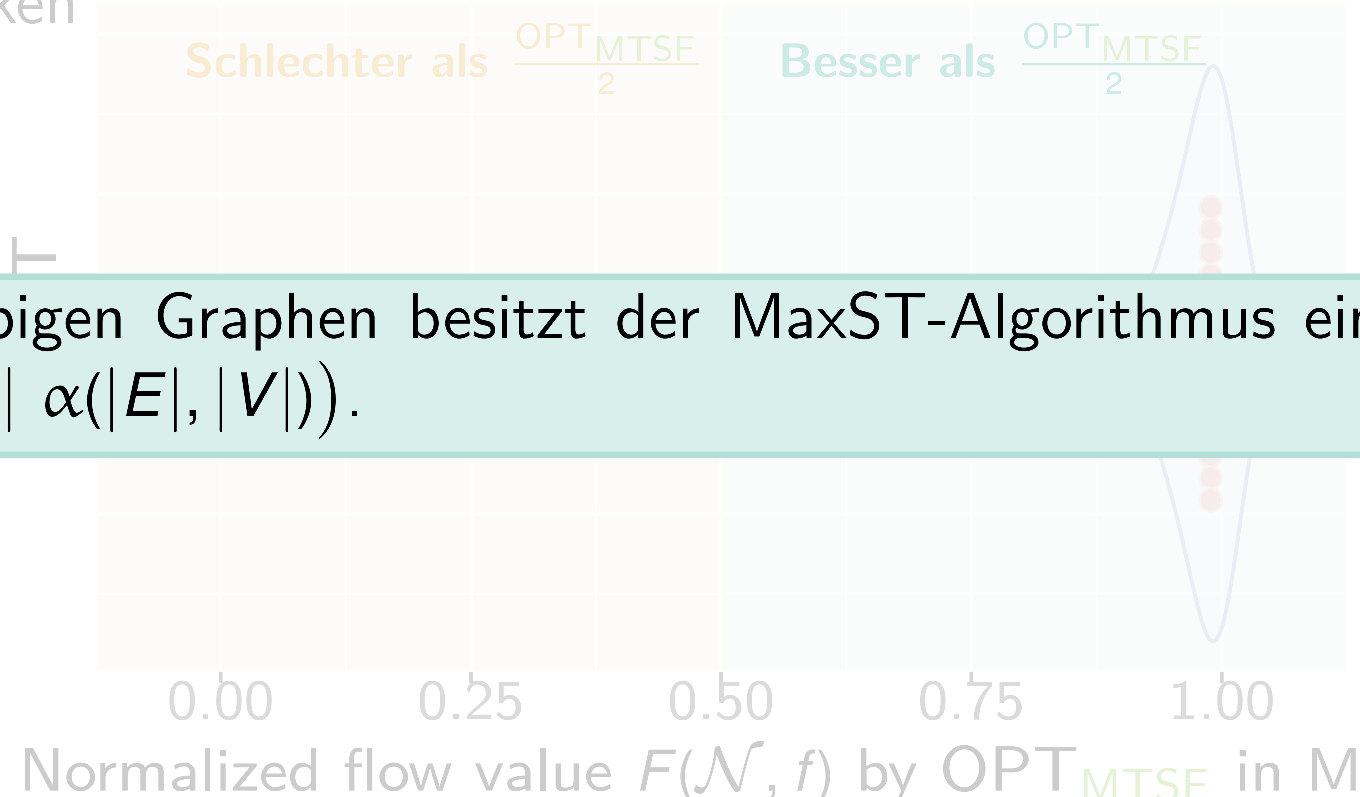
- Simulationen auf dem NESTA Benchmarksets, die realistischer sind als die IEEE-Benchmarksets, z.B., bzgl. der thermischen Leitungsschranken



MaxST auf **beliebigen Graphen** ist in den meisten Fällen sehr nah an einer optimalen Lösung OPT_{MTSF}.

Simulationen [Seite 349; Grastien et al., 2018]

- Simulationen auf dem NESTA Benchmarksets, die realistischer sind als die IEEE-Benchmarksets, z.B., bzgl. der thermischen Leitungsschranken

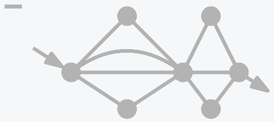
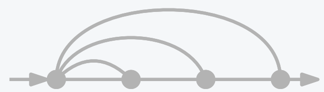
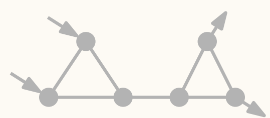
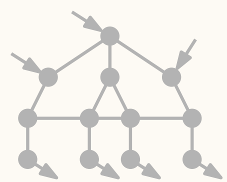
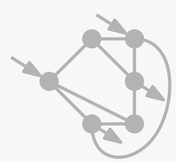
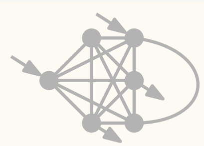


Auf beliebigen Graphen besitzt der MaxST-Algorithmus eine Laufzeit von $\mathcal{O}\big(|E|\ \alpha(|E|, |V|)\big)$.

MaxST auf **beliebigen Graphen** ist in den meisten Fällen sehr nah an einer optimalen Lösung $\mathrm{OPT}_{\mathrm{MTSF}}$.

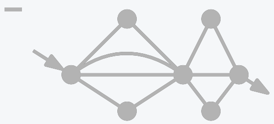
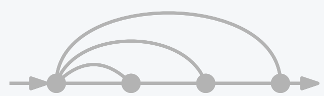
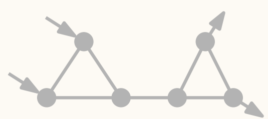
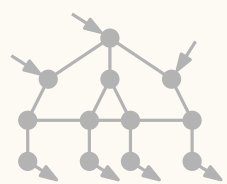
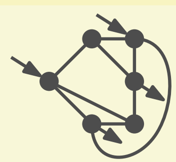
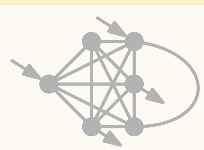
Überblick über MTSF Ergebnisse

Komplexität

	Graph Structure		Complexity	Algorithm
ein Erzeuger, ein Verbraucher	Penrose-Minoren-freie Graphen		Polynomialzeit lösbar	DTP ✓
	Serienparallele Graphen		NP-schwer	✗
beliebige Erzeuger, beliebige Verbraucher	Kakteen mit Maximalgrad 3		NP-schwer [Lehmann et al., 2014]	2-approx. ✓
	2-Level-Bäume		NP-schwer [Lehmann et al., 2014]	✗
	planar graphs with max degree of 3		strongly NP-hard [Lehmann et al., 2014]	✗
$\lvert V_G\rvert = 2$, $\lvert V_D\rvert = 2$	beliebige Graphen		nicht-APX [Lehmann et al., 2014]	✗

Überblick über MTSF Ergebnisse

Komplexität

	Graph Structure		Complexity	Algorithm
ein Erzeuger, ein Verbraucher	Penrose-Minoren-freie Graphen		Polynomialzeit lösbar	DTP ✓
	Serienparallele Graphen		NP-schwer	✗
beliebige Erzeuger, beliebige Verbraucher	Kakteen mit Maximalgrad 3		NP-schwer [Lehmann et al., 2014]	2-approx. ✓
	2-Level-Bäume		NP-schwer [Lehmann et al., 2014]	✗
	planare Graphen mit Maximalgrad 3		stark NP-schwer [Lehmann et al., 2014]	✗
$\lvert V_G \rvert = 2$, $\lvert V_D \rvert = 2$	beliebige Graphen		nicht-APX [Lehmann et al., 2014]	✗

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Hamilton Path Problem (HPP) & Hamilton Cycle Problem (HCP) [Seite 196; Skiena, 1990]

Hamilton Path Problem (HPP) [Seiten 199–200; Garey und Johnson, 1983]

Instanz: Ein Graph $G_{\text{HPP}} = (V_{\text{HPP}}, E_{\text{HPP}})$.

Frage: Existiert ein Pfad $\pi^\star \in \Pi$, der jeden Knoten genau einmal besucht.

Hamilton Cycle Problem (HCP)

Instanz: Ein Graph $G_{\text{HCP}} = (V_{\text{HCP}}, E_{\text{HCP}})$.

Frage: Existiert ein Graphkreis in G_{HCP}, der jeden Knoten genau einmal besucht.

- Ein Graph, der einen Hamiltonianpfad besitzt wird als *traceable Graph* bezeichnet.
- Hamiltoniankreis wird als Hamiltoniangraph bezeichnet
- HCP ist stark NP-vollständig selbst für planare und kubische Graphen

[Seite 95; Garey und Johnson, 1983]

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

MTSF auf planaren Graphen ist stark NP-schwer

[Lehmann et al., 2014]

Logische Folge 7 [Seite 13; Lehmann et al., 2014]

Das Zulässigkeitsproblem für planare Netzwerke mit einem maximalen Grad von 3 ist stark NP-schwer.

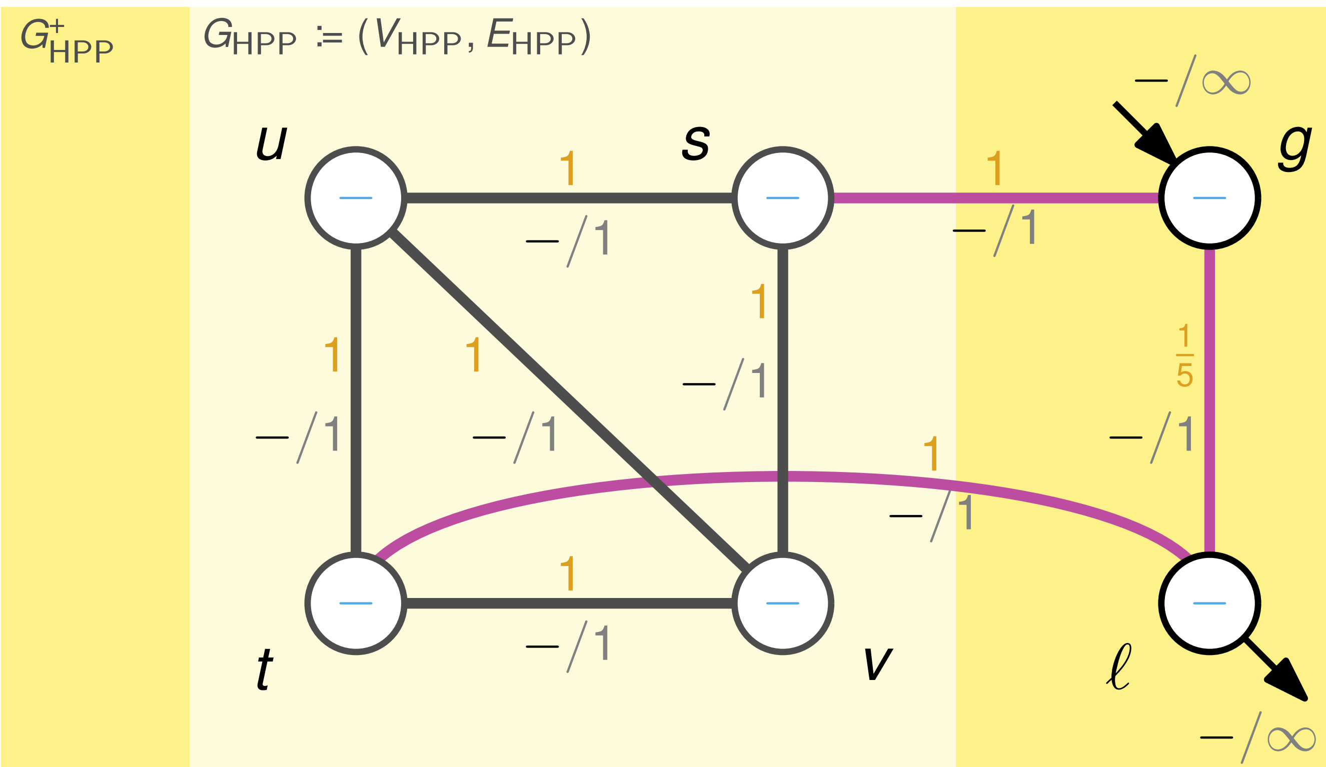
MTSF auf planaren Graphen ist stark NP-schwer

[Lehmann et al., 2014]

Karlsruher Institut für Technologie

Instanz $\mathcal{N} = (G = (V \cup V_{\text{HPP}}, E \cup E_{\text{HPP}}), V_G = \{g\}, V_D = \{\ell\}, \text{cap}, b, \overline{p_g}, \overline{p_d})$

- Knotenmenge $V_{\text{HPP}} = \{v_1, v_2, \ldots, v_n\}$, and $\overline{p_g} = \overline{p_d} = \infty$,
- Kantenmenge $E = E_{\text{HPP}} \cup \{(g, s), (t, \ell), (g, \ell)\}$,
- Kantenparameter $\text{cap}(e) \coloneqq b(e) \coloneqq 1 \quad \forall e \in E \setminus \{(g, \ell)\}$,
  $\text{cap}(g, \ell) \coloneqq 1, b(g, \ell) \coloneqq \frac{1}{n+1}$,

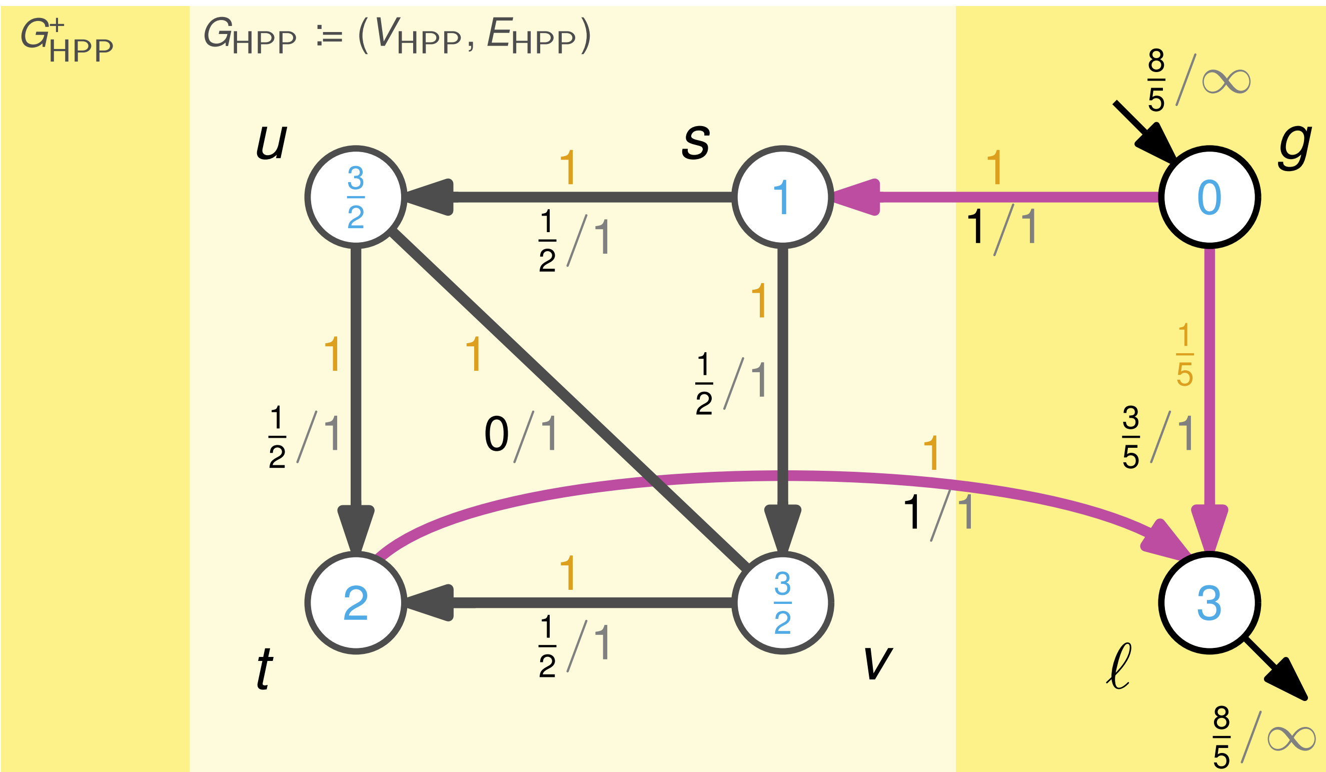


MTSF auf planaren Graphen ist stark NP-schwer

[Lehmann et al., 2014]

Instanz $\mathcal{N} = (G = (V \cup V_{\text{HPP}}, E \cup E_{\text{HPP}}), V_G = \{g\}, V_D = \{\ell\}, \text{cap}, b, \overline{p_g}, \overline{p_d})$

- Knotenmenge $V_{\text{HPP}} = \{v_1, v_2, \ldots, v_n\}$, and $\overline{p_g} = \overline{p_d} = \infty$,
- Kantenmenge $E = E_{\text{HPP}} \cup \{(g,s),(t,\ell),(g,\ell)\}$,
- Kantenparameter $\text{cap}(e) \coloneqq b(e) \coloneqq 1 \quad \forall e \in E \setminus \{(g,\ell)\}$,
  $\text{cap}(g,\ell) \coloneqq 1, b(g,\ell) \coloneqq \frac{1}{n+1}$,



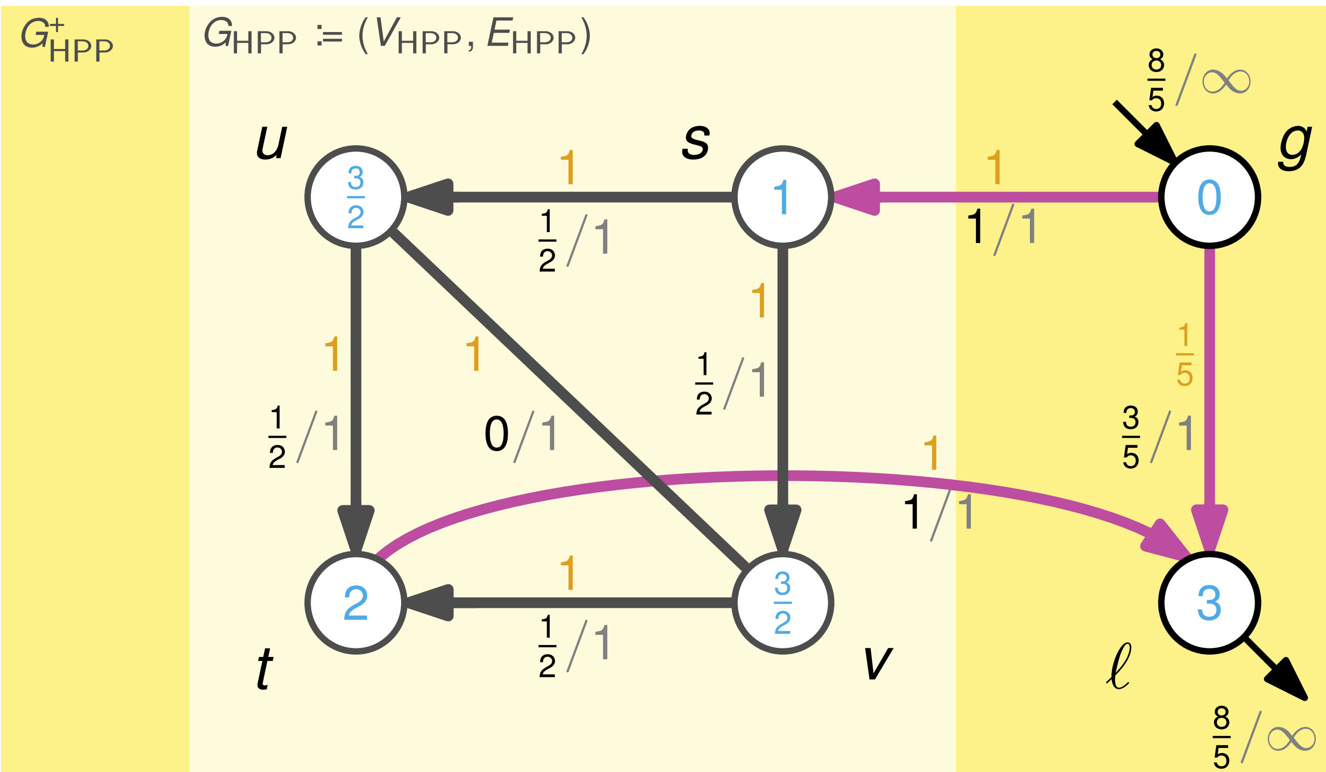
$\text{OPT}_{\text{MPF}}(\mathcal{N}) = \frac{8}{5}$

MTSF auf planaren Graphen ist stark NP-schwer

[Lehmann et al., 2014]

Instanz $\mathcal{N} = (G = (V \cup V_{\text{HPP}}, E \cup E_{\text{HPP}}), V_G = \{g\}, V_D = \{\ell\}, \text{cap}, b, \overline{p_g}, \overline{p_d})$

- Knotenmenge $V_{\text{HPP}} = \{v_1, v_2, \ldots, v_n\}$, and $\overline{p_g} = \overline{p_d} = \infty$,
- Kantenmenge $E = E_{\text{HPP}} \cup \{(g,s),(t,\ell),(g,\ell)\}$,
- Kantenparameter $\text{cap}(e) \coloneqq b(e) \coloneqq 1 \quad \forall e \in E \setminus \{(g,\ell)\}$,
  $\text{cap}(g,\ell) \coloneqq 1, b(g,\ell) \coloneqq \frac{1}{n+1}$,



G_{HPP} beschränkt den max $\Delta\theta^V(s,t)$, der den maximalen Leistungsfluss in $\mathcal{N}$ beschränkt

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

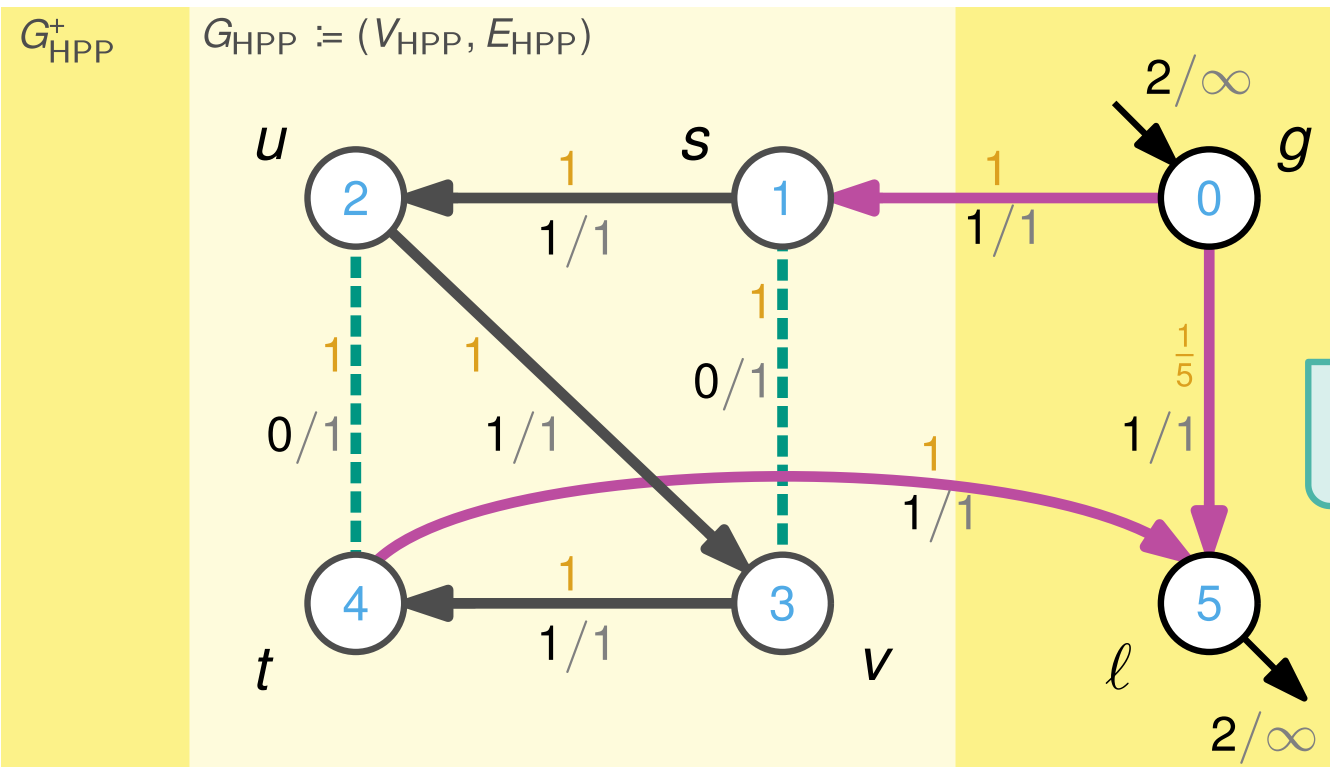
Institut für Theoretische Informatik
Lehrstuhl Algorithmik

MTSF auf planaren Graphen ist stark NP-schwer

[Lehmann et al., 2014]

Instanz $\mathcal{N} = (G = (V \cup V_{\text{HPP}}, E \cup E_{\text{HPP}}), V_G = \{g\}, V_D = \{\ell\}, \text{cap}, b, \overline{p_g}, \overline{p_d})$

- Knotenmenge $V_{\text{HPP}} = \{v_1, v_2, \ldots, v_n\}$, and $\overline{p_g} = \overline{p_d} = \infty$,
- Kantenmenge $E = E_{\text{HPP}} \cup \{(g,s),(t,\ell),(g,\ell)\}$,
- Kantenparameter $\text{cap}(e) \coloneqq b(e) \coloneqq 1 \quad \forall e \in E \setminus \{(g,\ell)\}$,
  $\text{cap}(g,\ell) \coloneqq 1,\ b(g,\ell) \coloneqq \frac{1}{n+1}$,



OPT$_{\text{MTSF}}(\mathcal{N}) = 2$ gdw. G_{HPP} einen Hamiltonpfad besitzt. ✓

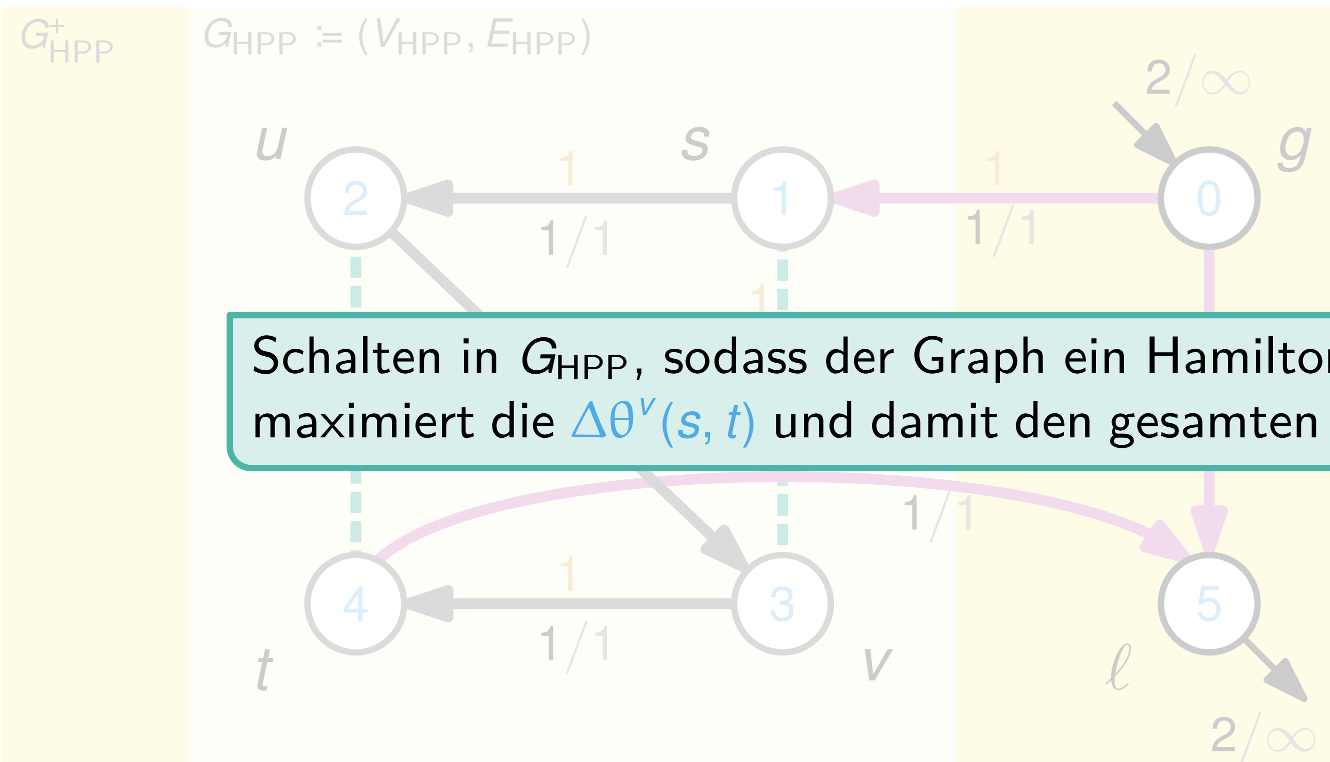
$\text{OPT}_{\text{MTSF}}(G^{+}_{\text{HPP}}) = 2$

MTSF auf planaren Graphen ist stark NP-schwer

[Lehmann et al., 2014]

Instanz $\mathcal{N} = (G = (V \cup V_{\text{HPP}}, E \cup E_{\text{HPP}}), V_G = \{g\}, V_D = \{\ell\}, \text{cap}, b, \overline{p_g}, \overline{p_d})$

- Knotenmenge $V_{\text{HPP}} = \{v_1, v_2, \ldots, v_n\}$, and $\overline{p_g} = \overline{p_d} = \infty$,
- Kantenmenge $E = E_{\text{HPP}} \cup \{(g, s), (t, \ell), (g, \ell)\}$,
- Kantenparameter $\text{cap}(e) \coloneqq b(e) \coloneqq 1 \quad \forall e \in E \setminus \{(g, \ell)\}$,
  $\text{cap}(g, \ell) \coloneqq 1, b(g, \ell) \coloneqq \frac{1}{n+1}$,

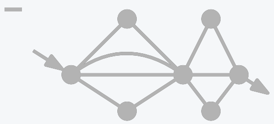
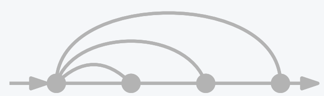
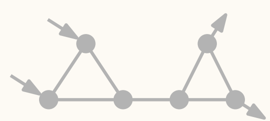
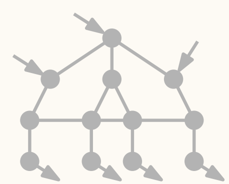
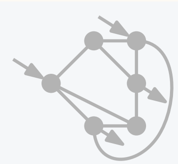
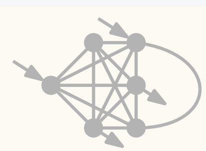


Schalten in G_{HPP}, sodass der Graph ein Hamiltoniangraph wird maximiert die $\Delta\theta^v(s, t)$ und damit den gesamten Leistungsfluss.

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

Überblick über die MTSF Ergebnisse

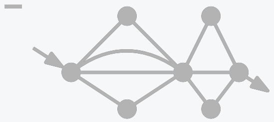
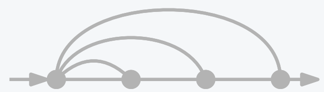
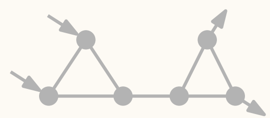
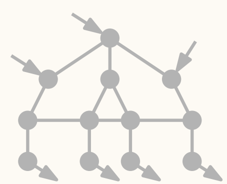
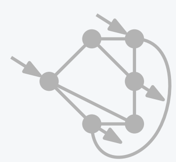
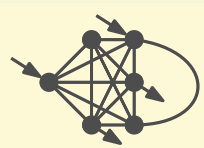
Komplexität ↓

	Graphenstruktur		Komplexität	Algorithmus
ein Erzeuger, ein Verbraucher	Penrose-Minoren-freie Graphen		Polynomialzeit lösbar	DTP ✓
	Serienparallele Graphen		NP-schwer	✗
beliebige Erzeuger, beliebige Verbraucher	Kakteen mit Maximalgrad 3		NP-schwer [Lehmann et al., 2014]	2-approx. ✓
	2-Level-Bäume		NP-schwer [Lehmann et al., 2014]	✗
	Planare Graphen mit Maximalgrad 3		stark NP-schwer [Lehmann et al., 2014]	✗
$\lvert V_G \rvert = 2$, $\lvert V_D \rvert = 2$	Beliebige Graphen		nicht-APX [Lehmann et al., 2014]	✗

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Überblick über die MTSF Ergebnisse

Komplexität

	Graphenstruktur		Komplexität	Algorithmus
ein Erzeuger, ein Verbraucher	Penrose-Minoren-freie Graphen		Polynomialzeit lösbar	DTP ✓
	Serienparallele Graphen		NP-schwer	✗
beliebige Erzeuger, beliebige Verbraucher	Kakteen mit Maximalgrad 3		NP-schwer [Lehmann et al., 2014]	2-approx. ✓
	2-Level-Bäume		NP-schwer [Lehmann et al., 2014]	✗
	Planare Graphen mit Maximalgrad 3		stark NP-schwer [Lehmann et al., 2014]	✗
$\|V_G\|=2$, $\|V_D\|=2$	Beliebige Graphen		nicht-APX [Lehmann et al., 2014]	✗

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Längste Pfade (LP) [Karger et al., 1997]

Karlsruher Institut für Technologie

Längste Pfade (LP)

Instanz: Ein Netzwerk $\mathcal{N}_{\text{LP}} = (G_{\text{LP}} = (V_{\text{LP}}, E_{\text{LP}}), \text{len})$, eine Längenfunktion $\text{len} \colon E \to \mathbb{R}_{\geq 0}$ und zwei ausgezeichnete Knoten $s, t \in V$.

Frage: Finde einen einfachen Pfad $\pi^\star \in \Pi(s,t)$ von s nach t von maximaler Länge—bedeutet $\max_{\pi \in \Pi(s,t)} \sum_{(u,v) \in \pi} \text{len}(u,v)$—in dem Netzwerk $\mathcal{N}_{\text{LP}}$.

- Auch als TAXICAB RIPOFF Problem bekannt

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Längste Pfade (LP) [Karger et al., 1997]

Längste Pfade (LP)

Instanz: Ein Netzwerk $\mathcal{N}_{\mathrm{LP}} = (G_{\mathrm{LP}} = (V_{\mathrm{LP}}, E_{\mathrm{LP}}), \mathrm{len})$, eine Längenfunktion $\mathrm{len}\colon E \to \mathbb{R}_{\geq 0}$ und zwei ausgezeichnete Knoten $s, t \in V$.

Frage: Finde einen einfachen Pfad $\pi^\star \in \Pi(s,t)$ von s nach t von maximaler Länge—bedeutet $\max_{\pi \in \Pi(s,t)} \sum_{(u,v) \in \pi} \mathrm{len}(u,v)$—in dem Netzwerk $\mathcal{N}_{\mathrm{LP}}$.

- Auch als TAXICAB RIPOFF Problem bekannt
- Für alle $\epsilon > 0$ ist es nicht möglich LP innerhalb eines Faktors von $2^{(\log n)^{1-\epsilon}}$ zu approximieren außer NP ist in quasipolynomial-deterministischer Zeit.

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

MTSF auf beliebigen Graphen ist nicht-APX

[Lehmann et al., 2014]

Karlsruher Institut für Technologie

Satz 8 [Seiten 10–12; Lehmann et al., 2014]

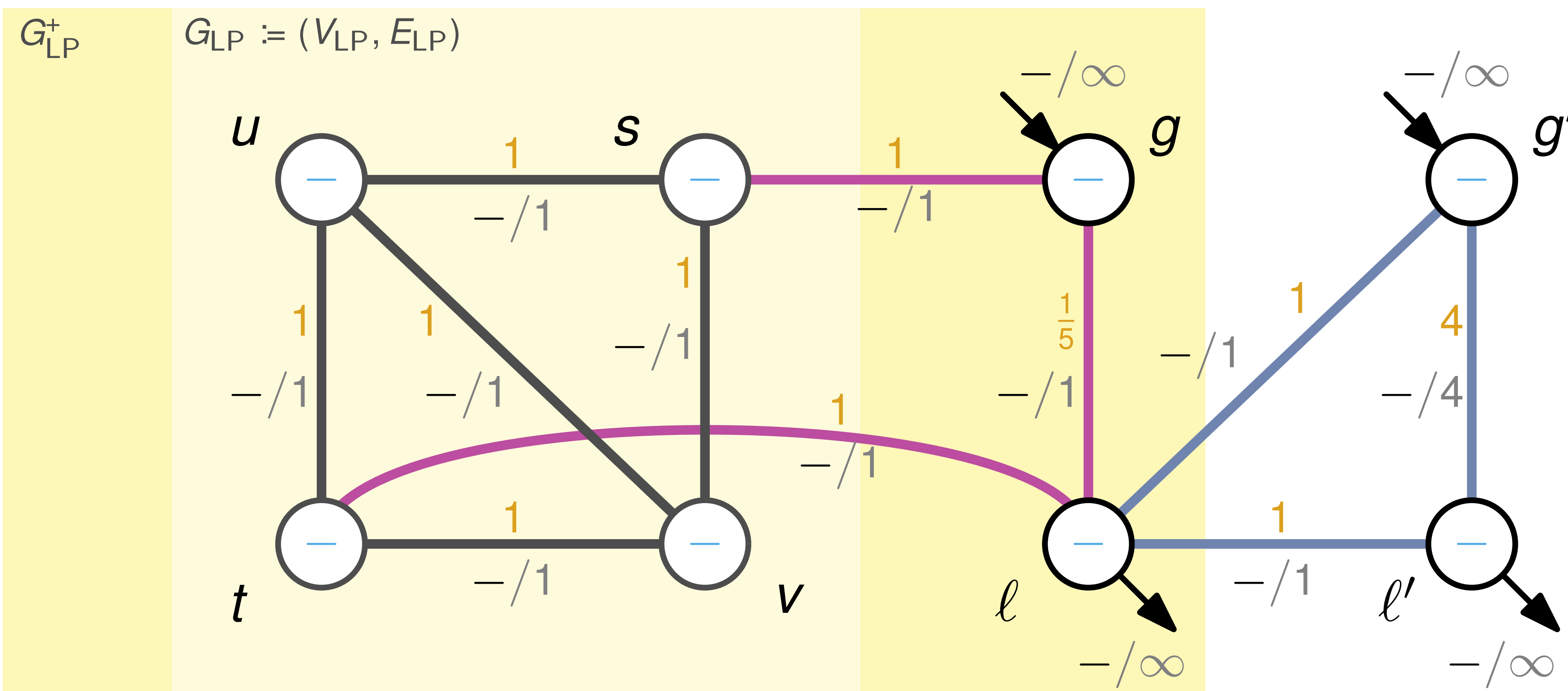
Es ist nicht möglich das MTSF-Problem innerhalb eines Faktors von $2^{(\log n)^{1-\epsilon}}$ zu approximieren außer NP ist in quasipolynomial-deterministischer Zeit enthalten.

MTSF auf beliebigen Graphen ist nicht-APX

[Lehmann et al., 2014]

Instanz $\mathcal{N} = (G = (V \cup V_{\text{LP}}, E \cup E_{\text{LP}}), V_G = \{g, g'\}, V_D = \{\ell, \ell'\}, \text{cap}, b, \overline{p_g}, \underline{p_{d_\ell}} = 3, \overline{p_d})$

- Knotenmenge $V_{\text{LP}} = \{v_1, v_2, \ldots, v_n\}$, und $\overline{p_g} = \overline{p_d} = \infty$,
- Kantenmenge $E = E_{\text{LP}} \cup \{(g, s), (t, \ell), (g, \ell), (g', \ell'), (g', \ell), (\ell', \ell)\}$,
- Kantenparameter $\text{cap}(e) \coloneqq b(e) \coloneqq 1 \quad \forall e \in E \setminus \{(g, \ell), (g', \ell')\}$,
  $\text{cap}(g, \ell) \coloneqq 1, b(g, \ell) \coloneqq \frac{1}{n+1}$,
  $\text{cap}(g', \ell') \coloneqq n, b(g', \ell') \coloneqq n$

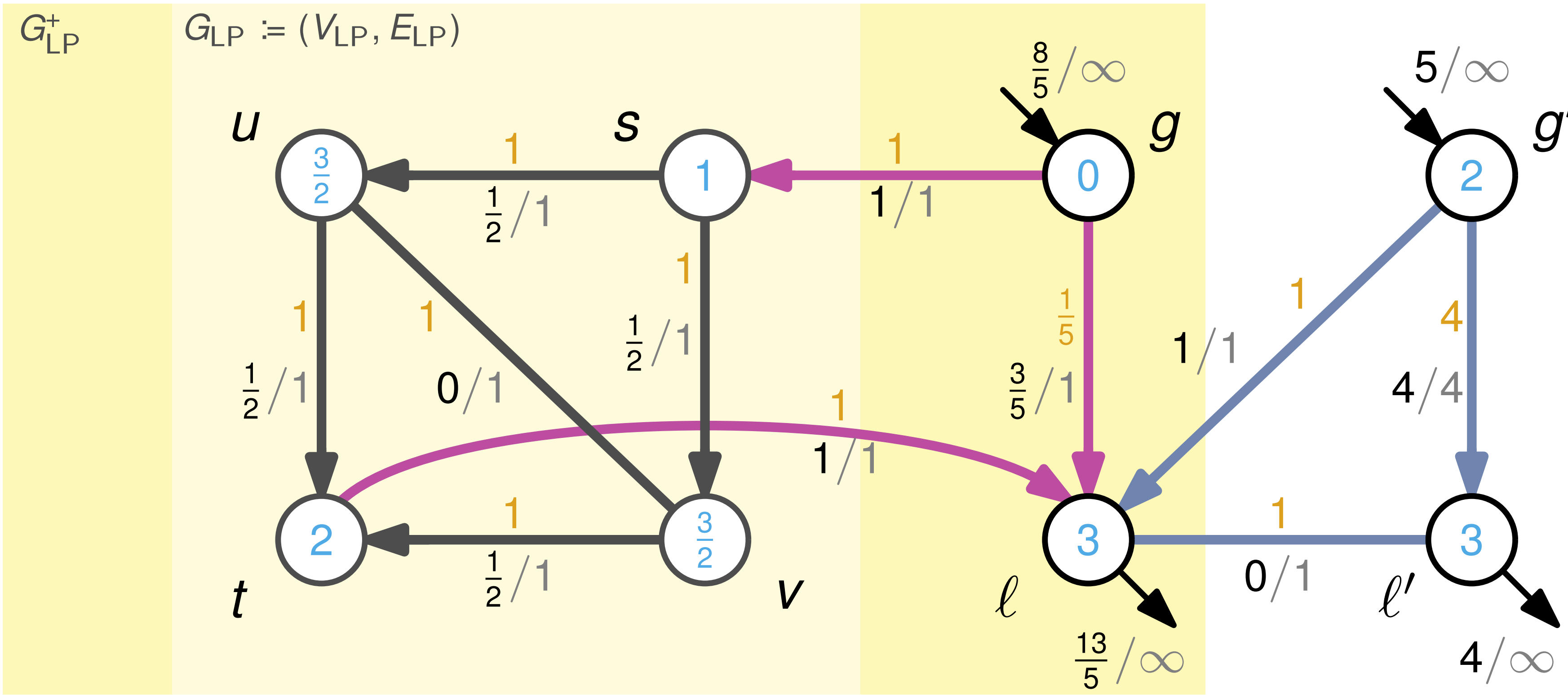


MTSF auf beliebigen Graphen ist nicht-APX

[Lehmann et al., 2014]

Instanz $\mathcal{N} = (G = (V \cup V_{\text{LP}}, E \cup E_{\text{LP}}), V_G = \{g, g'\}, V_D = \{\ell, \ell'\}, \text{cap}, b, \overline{p_g}, \underline{p_{d_\ell}} = 3, \overline{p_d})$

- Knotenmenge $V_{\text{LP}} = \{v_1, v_2, \dots, v_n\}$, und $\overline{p_g} = \overline{p_d} = \infty$,
- Kantenmenge $E = E_{\text{LP}} \cup \{(g, s), (t, \ell), (g, \ell), (g', \ell'), (g', \ell), (\ell', \ell)\}$,
- Kantenparameter $\text{cap}(e) \coloneqq b(e) \coloneqq 1 \quad \forall e \in E \setminus \{(g, \ell), (g', \ell')\}$,
  $\text{cap}(g, \ell) \coloneqq 1, b(g, \ell) \coloneqq \frac{1}{n+1}$,
  $\text{cap}(g', \ell') \coloneqq n, b(g', \ell') \coloneqq n$



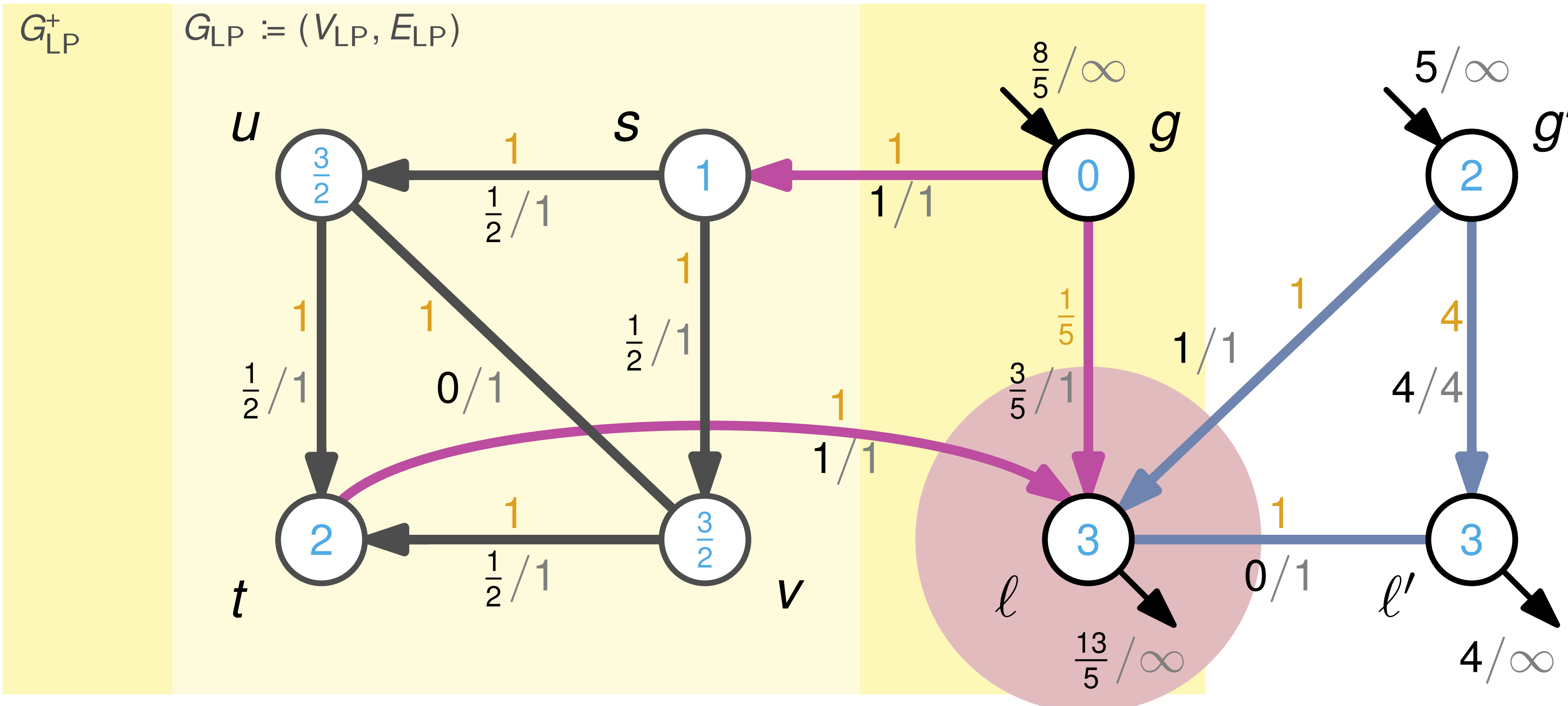
$\text{OPT}_{\text{MPFP}}(\mathcal{N}) = \frac{33}{5}$

MTSF auf beliebigen Graphen ist nicht-APX

[Lehmann et al., 2014]

Instanz $\mathcal{N} = (G = (V \cup V_{\text{LP}}, E \cup E_{\text{LP}}), V_G = \{g, g'\}, V_D = \{\ell, \ell'\}, \text{cap}, b, \overline{p_g}, \overline{p_{d_\ell}} = 3, \overline{p_d})$ ⚡

- Knotenmenge $V_{\text{LP}} = \{v_1, v_2, \ldots, v_n\}$, und $\overline{p_g} = \overline{p_d} = \infty$,
- Kantenmenge $E = E_{\text{LP}} \cup \{(g,s),(t,\ell),(g,\ell),(g',\ell'),(g',\ell),(\ell',\ell)\}$,
- Kantenparameter $\text{cap}(e) \coloneqq b(e) \coloneqq 1 \quad \forall e \in E \setminus \{(g,\ell),(g',\ell')\}$,
  $\text{cap}(g,\ell) \coloneqq 1, b(g,\ell) \coloneqq \frac{1}{n+1}$,
  $\text{cap}(g',\ell') \coloneqq n, b(g',\ell') \coloneqq n$



$3 \leq \frac{13}{5} \leq \infty$ ⚡

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

MTSF auf beliebigen Graphen ist nicht-APX

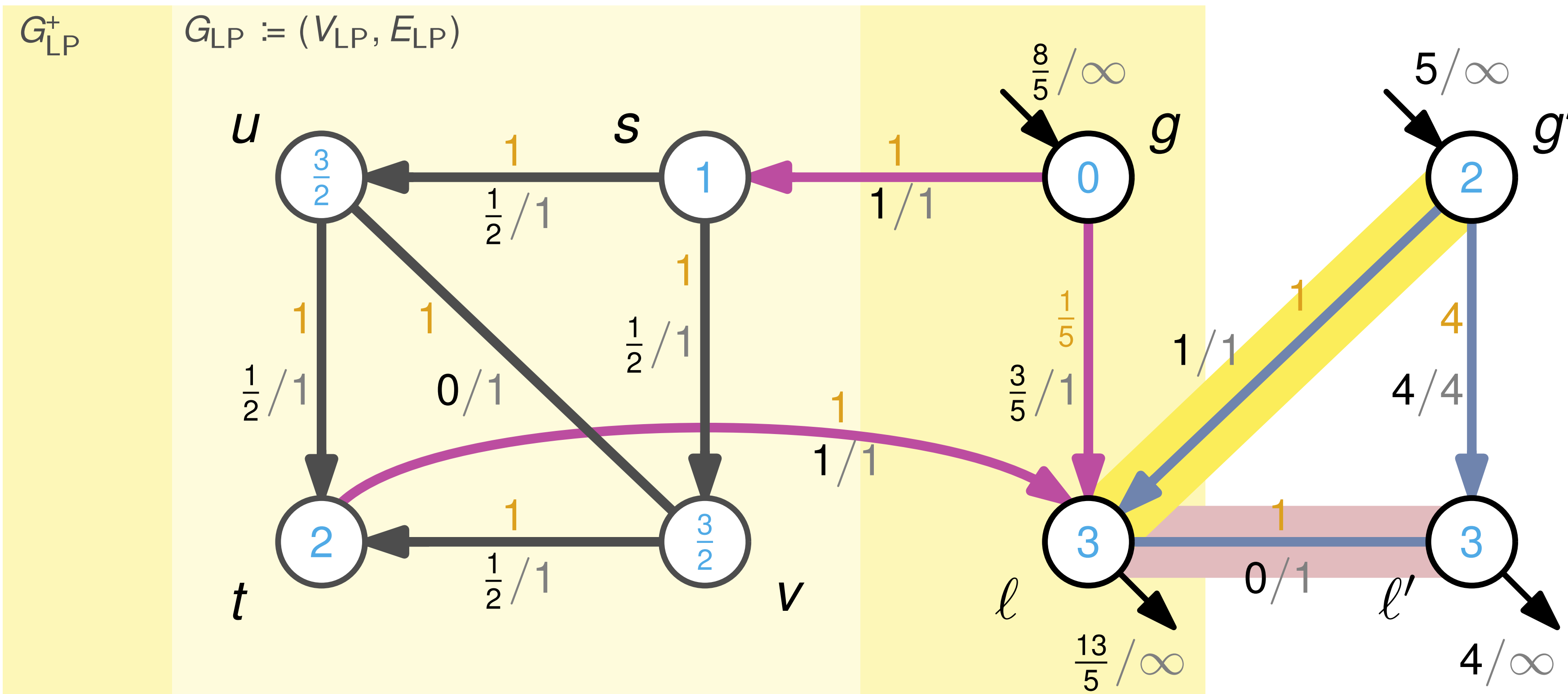
[Lehmann et al., 2014]

Instanz $\mathcal{N} = (G = (V \cup V_{\mathrm{LP}}, E \cup E_{\mathrm{LP}}), V_G = \{g, g'\}, V_D = \{\ell, \ell'\}, \mathrm{cap}, b, \overline{p_g}, \underline{p_{d_\ell}} = 3, \overline{p_d})$

- Knotenmenge $V_{\mathrm{LP}} = \{v_1, v_2, \dots, v_n\}$, und $\overline{p_g} = \overline{p_d} = \infty$,
- Kantenmenge $E = E_{\mathrm{LP}} \cup \{(g, s), (t, \ell), (g, \ell), (g', \ell'), (g', \ell), (\ell', \ell)\}$,
- Kantenparameter $\mathrm{cap}(e) \coloneqq b(e) \coloneqq 1 \quad \forall e \in E \setminus \{(g, \ell), (g', \ell')\}$,

  $\mathrm{cap}(g, \ell) \coloneqq 1, b(g, \ell) \coloneqq \frac{1}{n+1}$,

  $\mathrm{cap}(g', \ell') \coloneqq n, b(g', \ell') \coloneqq n$



$\mathrm{OPT}_{\mathrm{MPFP}}(\mathcal{N}) = \frac{33}{5}$

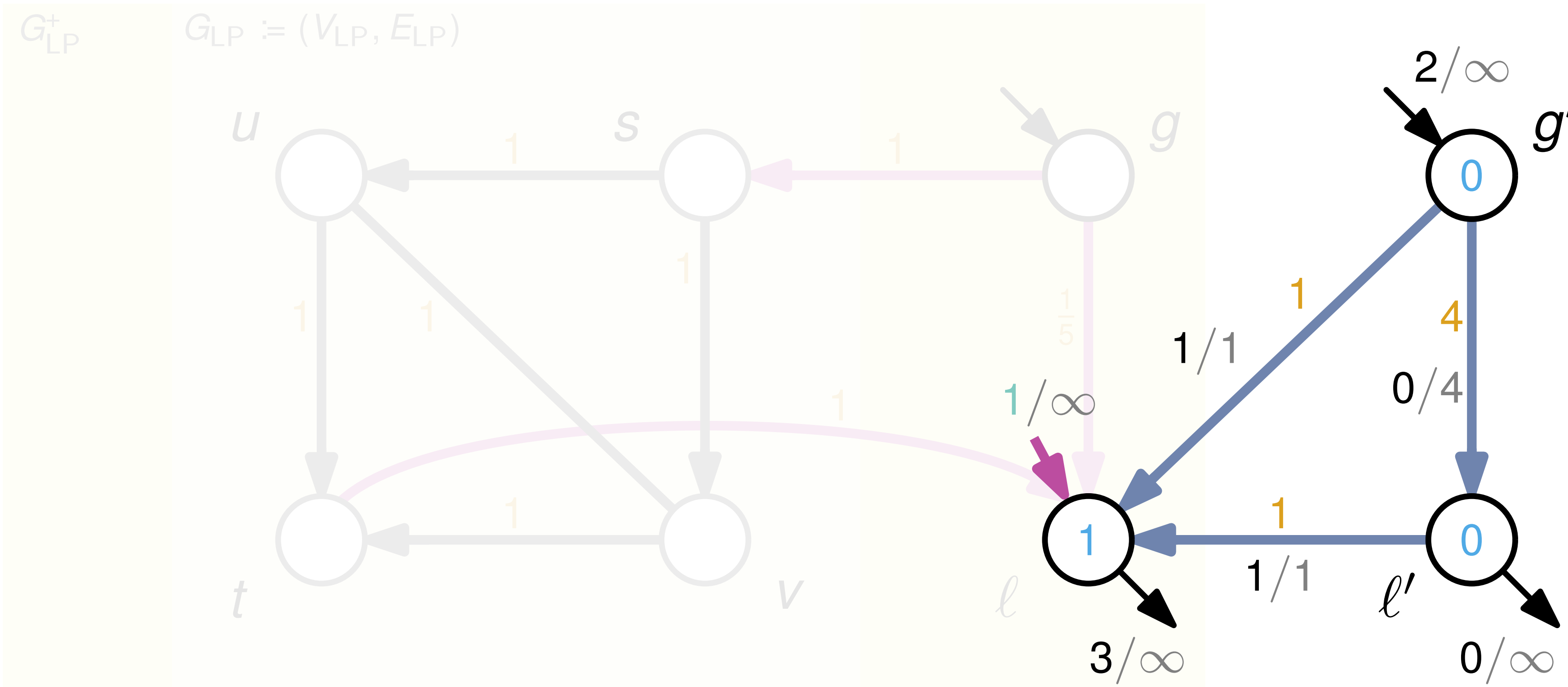
Institut für Theoretische Informatik
Lehrstuhl Algorithmik

MTSF auf beliebigen Graphen ist nicht-APX

[Lehmann et al., 2014]

Instanz $\mathcal{N} = (G = (V \cup V_{\text{LP}}, E \cup E_{\text{LP}}), V_G = \{g, g'\}, V_D = \{\ell, \ell'\}, \text{cap}, b, \overline{p_g}, \underline{p_d}_{\ell} = 3, \overline{p_d})$

- Knotenmenge $V_{\text{LP}} = \{v_1, v_2, \ldots, v_n\}$, und $\overline{p_g} = \overline{p_d} = \infty$,
- Kantenmenge $E = E_{\text{LP}} \cup \{(g, s), (t, \ell), (g, \ell), (g', \ell'), (g', \ell), (\ell', \ell)\}$,
- Kantenparameter $\text{cap}(e) \coloneqq b(e) \coloneqq 1 \quad \forall e \in E \setminus \{(g, \ell), (g', \ell')\}$,
  $\text{cap}(g, \ell) \coloneqq 1, b(g, \ell) \coloneqq \frac{1}{n+1}$,
  $\text{cap}(g', \ell') \coloneqq n, b(g', \ell') \coloneqq n$



Institut für Theoretische Informatik
Lehrstuhl Algorithmik

MTSF auf beliebigen Graphen ist nicht-APX

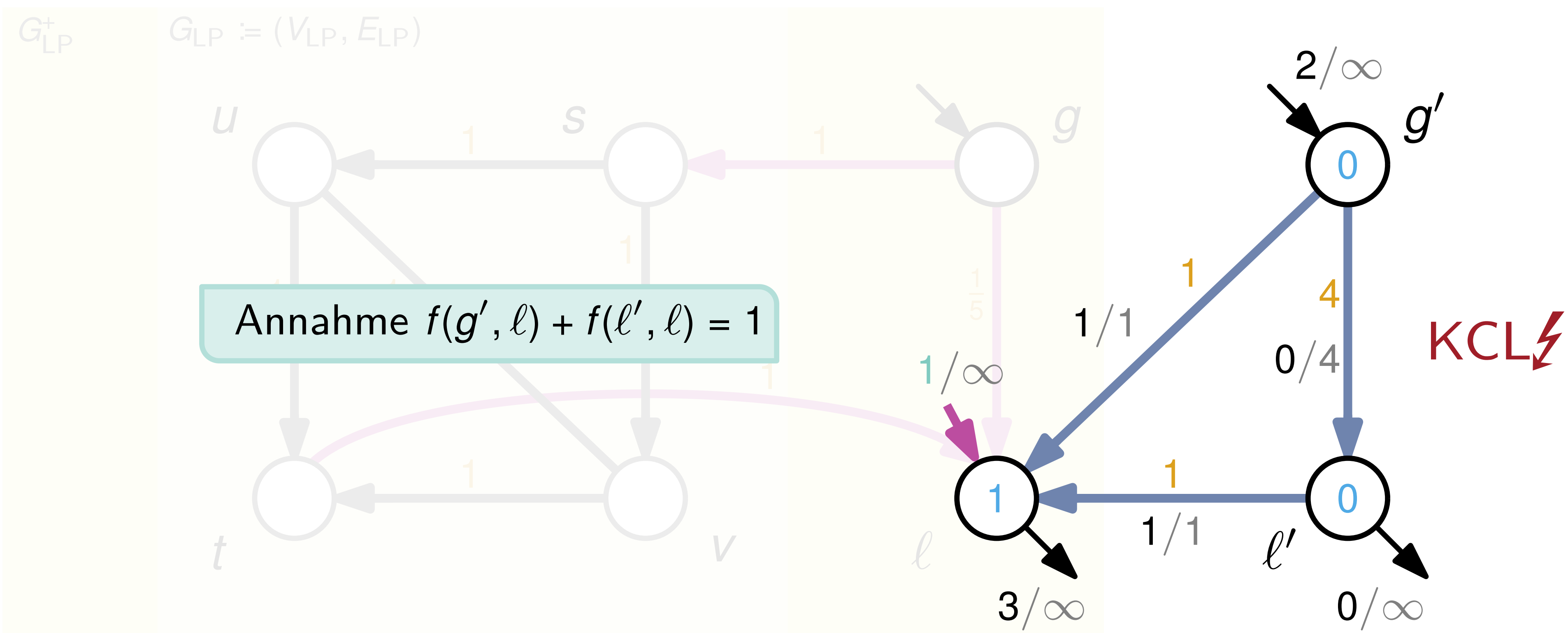
[Lehmann et al., 2014]

Instanz $\mathcal{N} = (G = (V \cup V_{\text{LP}}, E \cup E_{\text{LP}}), V_G = \{g, g'\}, V_D = \{\ell, \ell'\}, \text{cap}, b, \overline{p_g}, \underline{p_d}_\ell = 3, \overline{p_d})$

- Knotenmenge $V_{\text{LP}} = \{v_1, v_2, \dots, v_n\}$, und $\overline{p_g} = \overline{p_d} = \infty$,
- Kantenmenge $E = E_{\text{LP}} \cup \{(g,s), (t,\ell), (g,\ell), (g',\ell'), (g',\ell), (\ell',\ell)\}$,
- Kantenparameter $\text{cap}(e) \coloneqq b(e) \coloneqq 1 \quad \forall e \in E \setminus \{(g,\ell), (g',\ell')\}$,

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

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



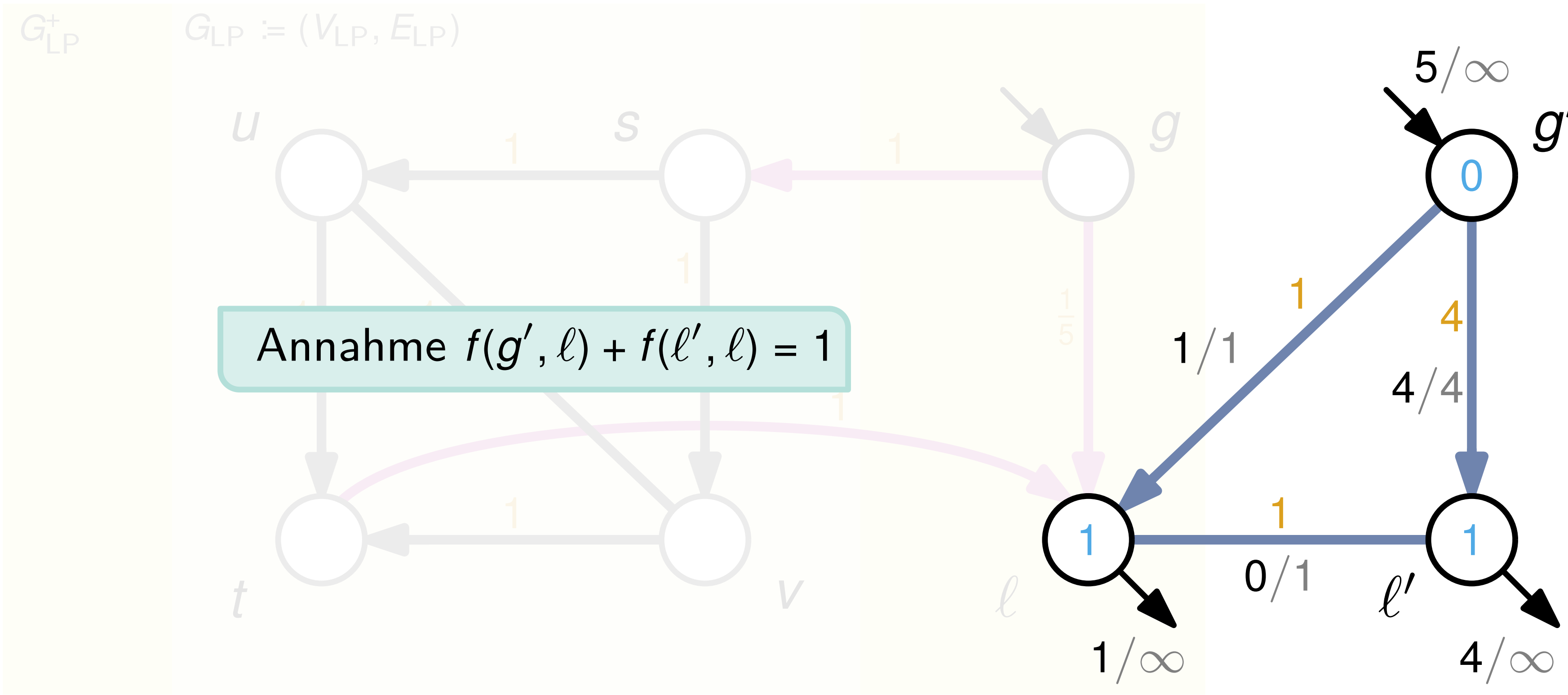
Was ist der maximal mögliche Leistungsfluss in $(\mathcal{N} \setminus G_{\text{LP}}) - g$?

MTSF auf beliebigen Graphen ist nicht-APX

[Lehmann et al., 2014]

Instanz $\mathcal{N} = (G = (V \cup V_{\text{LP}}, E \cup E_{\text{LP}}), V_G = \{g, g'\}, V_D = \{\ell, \ell'\}, \text{cap}, b, \overline{p_g}, \underline{p_{d_\ell}} = 3, \overline{p_d})$

- Knotenmenge $V_{\text{LP}} = \{v_1, v_2, \ldots, v_n\}$, und $\overline{p_g} = \overline{p_d} = \infty$,
- Kantenmenge $E = E_{\text{LP}} \cup \{(g,s),(t,\ell),(g,\ell),(g',\ell'),(g',\ell),(\ell',\ell)\}$,
- Kantenparameter $\text{cap}(e) \coloneqq b(e) \coloneqq 1 \quad \forall e \in E \setminus \{(g,\ell),(g',\ell')\}$,
  $\text{cap}(g,\ell) \coloneqq 1, b(g,\ell) \coloneqq \frac{1}{n+1}$,
  $\text{cap}(g',\ell') \coloneqq n, b(g',\ell') \coloneqq n$



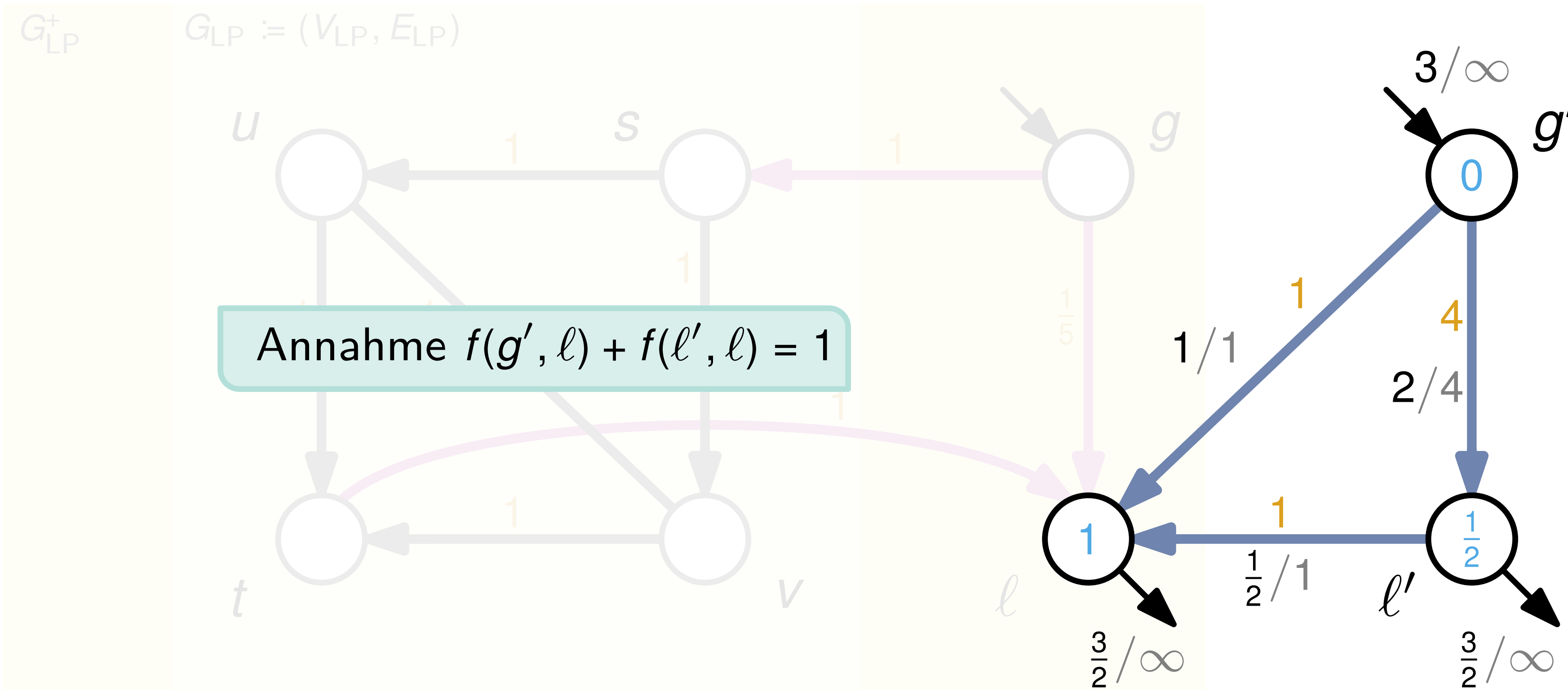
Was ist der maximal mögliche Leistungsfluss in $(\mathcal{N} \setminus G_{\text{LP}}) - g$? $\text{OPT}_{\text{MPFP}}((\mathcal{N} \setminus G_{\text{LP}}) - g) = 5$

MTSF auf beliebigen Graphen ist nicht-APX

[Lehmann et al., 2014]

Instanz $\mathcal{N} = (G = (V \cup V_{\text{LP}}, E \cup E_{\text{LP}}), V_G = \{g, g'\}, V_D = \{\ell, \ell'\}, \text{cap}, b, \overline{p_g}, \underline{p_d}_{\ell} = 3, \overline{p_d})$

- Knotenmenge $V_{\text{LP}} = \{v_1, v_2, \dots, v_n\}$, und $\overline{p_g} = \overline{p_d} = \infty$,
- Kantenmenge $E = E_{\text{LP}} \cup \{(g,s), (t,\ell), (g,\ell), (g',\ell'), (g',\ell), (\ell',\ell)\}$,
- Kantenparameter $\text{cap}(e) \coloneqq b(e) \coloneqq 1 \quad \forall e \in E \setminus \{(g,\ell), (g',\ell')\}$,
  $\text{cap}(g,\ell) \coloneqq 1, b(g,\ell) \coloneqq \frac{1}{n+1}$,
  $\text{cap}(g',\ell') \coloneqq n, b(g',\ell') \coloneqq n$



Was ist der maximal mögliche Verbrauch für ℓ in $(\mathcal{N} \setminus G_{\text{LP}}) - g$?

— $\max f_{\text{net}}(\ell) = 3$

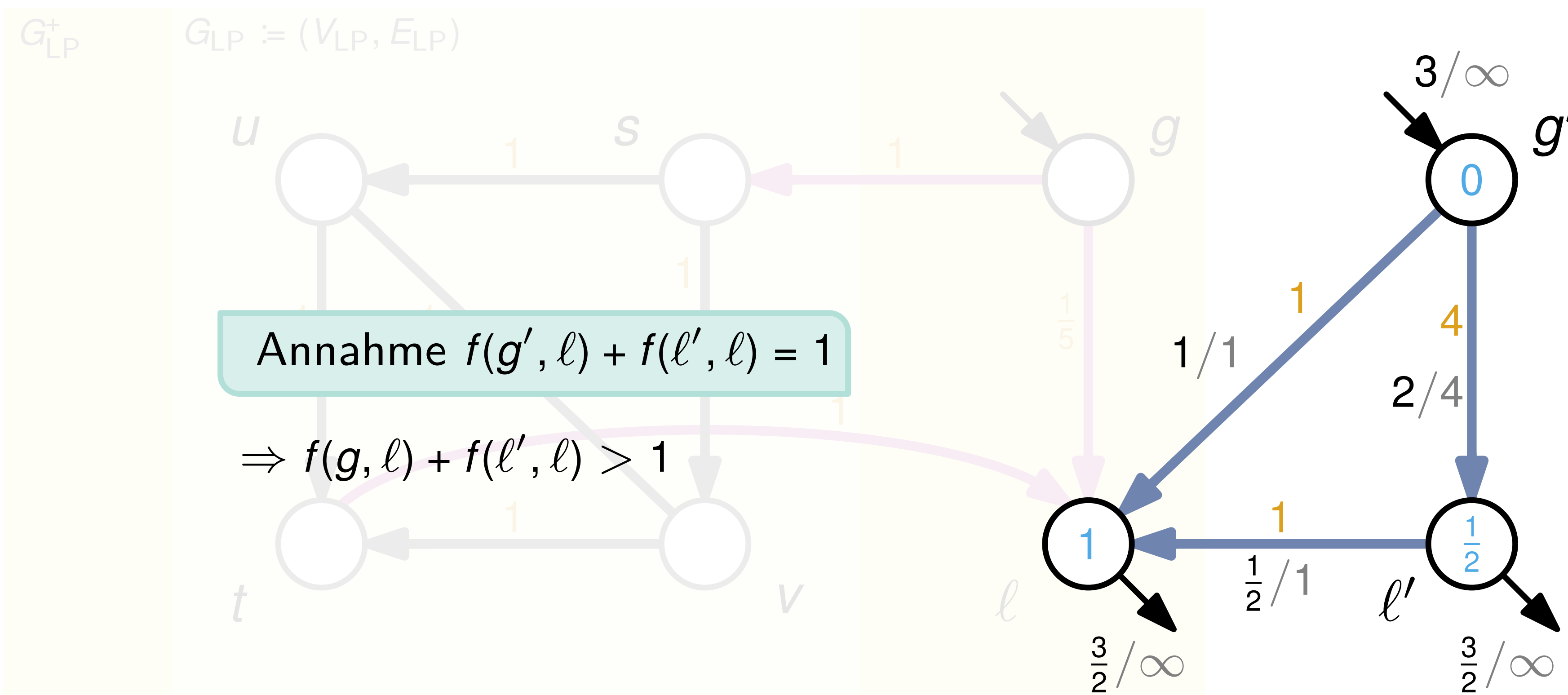
Institut für Theoretische Informatik
Lehrstuhl Algorithmik

MTSF auf beliebigen Graphen ist nicht-APX

[Lehmann et al., 2014]

Instanz $\mathcal{N} = (G = (V \cup V_{\text{LP}}, E \cup E_{\text{LP}}), V_G = \{g, g'\}, V_D = \{\ell, \ell'\}, \text{cap}, b, \overline{p_g}, \underline{p_d}_{\ell} = 3, \overline{p_d})$

- Knotenmenge $V_{\text{LP}} = \{v_1, v_2, \ldots, v_n\}$, und $\overline{p_g} = \overline{p_d} = \infty$,
- Kantenmenge $E = E_{\text{LP}} \cup \{(g,s), (t,\ell), (g,\ell), (g',\ell'), (g',\ell), (\ell',\ell)\}$,
- Kantenparameter $\text{cap}(e) \coloneqq b(e) \coloneqq 1 \quad \forall e \in E \setminus \{(g,\ell), (g',\ell')\}$,
  $\text{cap}(g,\ell) \coloneqq 1, b(g,\ell) \coloneqq \frac{1}{n+1}$,
  $\text{cap}(g',\ell') \coloneqq n, b(g',\ell') \coloneqq n$



Annahme $f(g',\ell) + f(\ell',\ell) = 1$

$\Rightarrow f(g,\ell) + f(\ell',\ell) > 1$

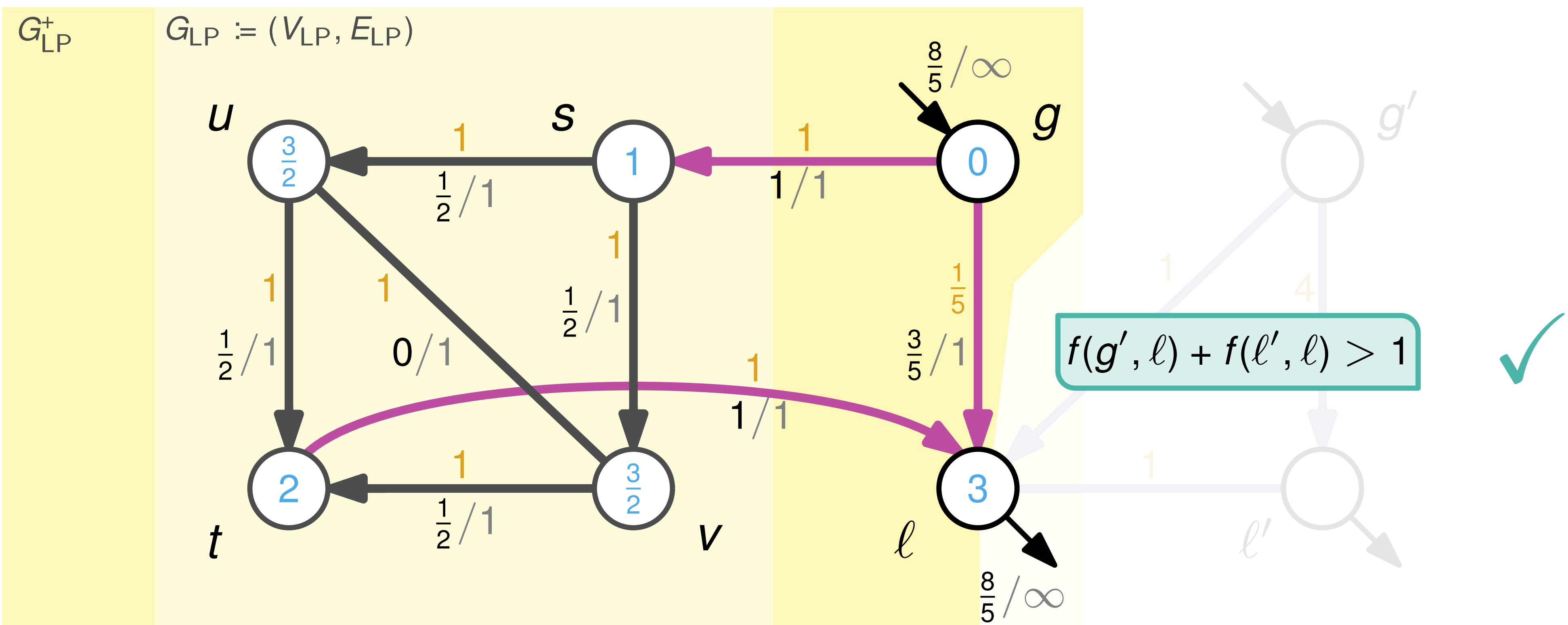
Was ist der maximal mögliche Verbrauch für ℓ in $(\mathcal{N} \setminus G_{\text{LP}}) - g$? — $\max f_{\text{net}}(\ell) = 3$

MTSF auf beliebigen Graphen ist nicht-APX

[Lehmann et al., 2014]

Instanz $\mathcal{N} = (G = (V \cup V_{LP}, E \cup E_{LP}), V_G = \{g, g'\}, V_D = \{\ell, \ell'\}, \text{cap}, b, \overline{p_g}, \underline{p_{d_\ell}} = 3, \overline{p_d})$

- Knotenmenge $V_{LP} = \{v_1, v_2, \ldots, v_n\}$, und $\overline{p_g} = \overline{p_d} = \infty$,
- Kantenmenge $E = E_{LP} \cup \{(g,s), (t,\ell), (g,\ell), (g',\ell'), (g',\ell), (\ell',\ell)\}$,
- Kantenparameter $\text{cap}(e) \coloneqq b(e) \coloneqq 1 \quad \forall e \in E \setminus \{(g,\ell), (g',\ell')\}$,
  $\text{cap}(g,\ell) \coloneqq 1, b(g,\ell) \coloneqq \frac{1}{n+1}$,
  $\text{cap}(g',\ell') \coloneqq n, b(g',\ell') \coloneqq n$



G_{LP} beschränkt die max $\Delta\theta^v(s,t)$, die den maximalen Leistungsfluss in $\mathcal{N}$ beschränkt

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

MTSF auf beliebigen Graphen ist nicht-APX

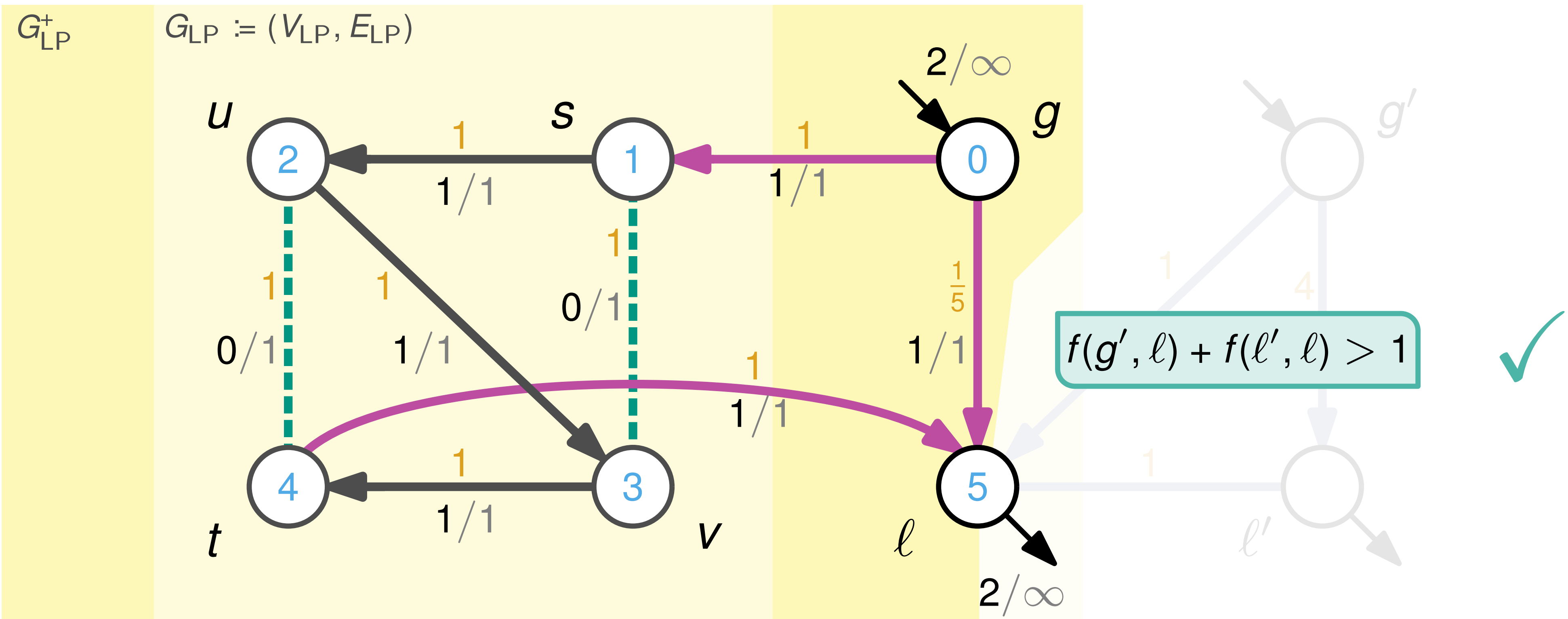
[Lehmann et al., 2014]

Instanz $\mathcal{N} = (G = (V \cup V_{LP}, E \cup E_{LP}), V_G = \{g, g'\}, V_D = \{\ell, \ell'\}, \text{cap}, b, \overline{p_g}, \underline{p_{d_\ell}} = 3, \overline{p_d})$

- Knotenmenge $V_{LP} = \{v_1, v_2, \ldots, v_n\}$, und $\overline{p_g} = \overline{p_d} = \infty$,
- Kantenmenge $E = E_{LP} \cup \{(g, s), (t, \ell), (g, \ell), (g', \ell'), (g', \ell), (\ell', \ell)\}$,
- Kantenparameter $\text{cap}(e) \coloneqq b(e) \coloneqq 1 \quad \forall e \in E \setminus \{(g, \ell), (g', \ell')\}$,

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

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



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

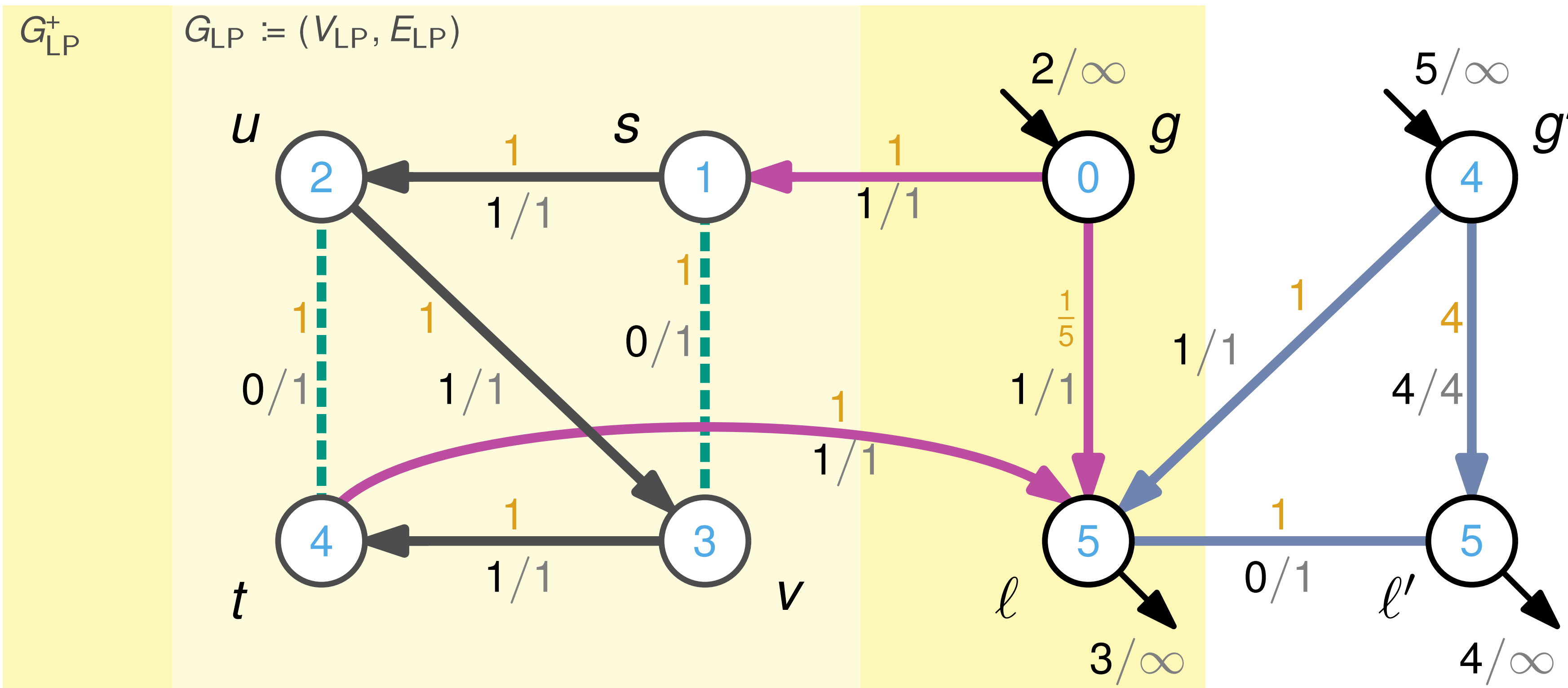
Institut für Theoretische Informatik
Lehrstuhl Algorithmik

MTSF auf beliebigen Graphen ist nicht-APX

[Lehmann et al., 2014]

Instanz $\mathcal{N} = (G = (V \cup V_{LP}, E \cup E_{LP}), V_G = \{g, g'\}, V_D = \{\ell, \ell'\}, \text{cap}, b, \overline{p_g}, \underline{p_d}_\ell = 3, \overline{p_d})$

- Knotenmenge $V_{LP} = \{v_1, v_2, \ldots, v_n\}$, und $\overline{p_g} = \overline{p_d} = \infty$,
- Kantenmenge $E = E_{LP} \cup \{(g, s), (t, \ell), (g, \ell), (g', \ell'), (g', \ell), (\ell', \ell)\}$,
- Kantenparameter $\text{cap}(e) \coloneqq b(e) \coloneqq 1 \quad \forall e \in E \setminus \{(g, \ell), (g', \ell')\}$,
  $\text{cap}(g, \ell) \coloneqq 1, b(g, \ell) \coloneqq \frac{1}{n+1}$,
  $\text{cap}(g', \ell') \coloneqq n, b(g', \ell') \coloneqq n$



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

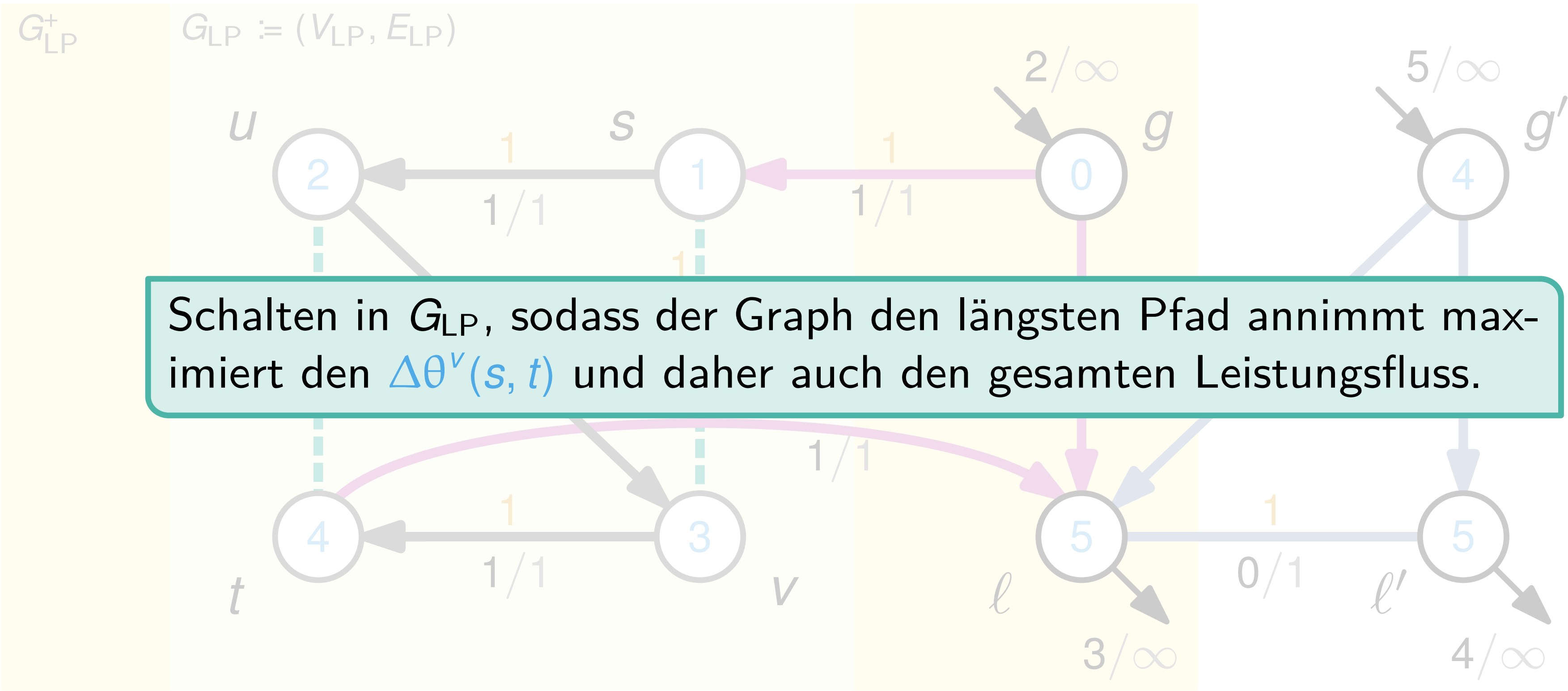
Institut für Theoretische Informatik
Lehrstuhl Algorithmik

$\textcolor{green}{\text{MTSF}}$ auf beliebigen Graphen ist nicht-APX

[Lehmann et al., 2014]

Instanz $\mathcal{N} = (G = (V \cup V_{\mathrm{LP}}, E \cup E_{\mathrm{LP}}), V_G = \{g, g'\}, V_D = \{\ell, \ell'\}, \mathrm{cap}, b, \overline{p_g}, \underline{p_{d_\ell}} = 3, \overline{p_d})$

- Knotenmenge $V_{\mathrm{LP}} = \{v_1, v_2, \ldots, v_n\}$, und $\overline{p_g} = \overline{p_d} = \infty$,
- Kantenmenge $E = E_{\mathrm{LP}} \cup \{(g,s), (t,\ell), (g,\ell), (g',\ell'), (g',\ell), (\ell',\ell)\}$,
- Kantenparameter $\mathrm{cap}(e) \coloneqq b(e) \coloneqq 1 \quad \forall e \in E \setminus \{(g,\ell), (g',\ell')\}$,
  $\mathrm{cap}(g,\ell) \coloneqq 1, b(g,\ell) \coloneqq \frac{1}{n+1}$,
  $\mathrm{cap}(g',\ell') \coloneqq n, b(g',\ell') \coloneqq n$

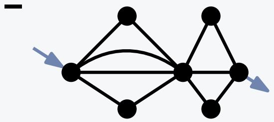
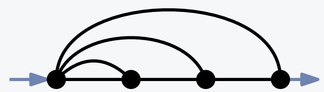
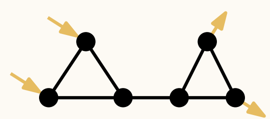
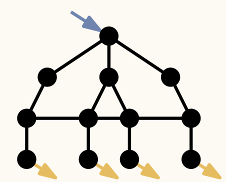
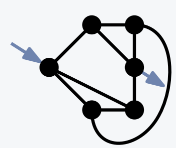
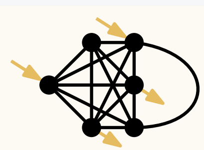


Schalten in G_{LP}, sodass der Graph den längsten Pfad annimmt maximiert den $\Delta\theta^v(s,t)$ und daher auch den gesamten Leistungsfluss.

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

Zusammenfassung & zukünftige Arbeiten

Komplexität

	Graphenstruktur		Komplexität	Algorithmen
ein Erzeuger, ein Verbraucher	Penrose-Minoren-freie Graphen		Polynomialzeit lösbar	✓
	Serienparallele Graphen		NP-schwer	✗
beliebige Erzeuger, beliebige Verbraucher	Kakteen mit Maximalgrad 3		NP-schwer [Lehmann et al., 2014]	✓
	2-Level-Bäume		NP-schwer [Lehmann et al., 2014]	✗
	Planare Graphen mit Maximalgrad 3		stark NP-schwer [Lehmann et al., 2014]	✗
$\lvert V_G\rvert=2$, $\lvert V_D\rvert=2$	Beliebige Graphen		nicht-APX [Lehmann et al., 2014]	✗

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Zusammenfassung & zukünftige Arbeiten

Graphenstruktur	Komplexität	Algorithmen
Penrose-Minoren-freie Graphen	Polynomialzeit lösbar	✓
Serienparallele Graphen	NP-schwer	✗
2-Level-Bäume	NP-schwer [Lehmann et al., 2014]	✗
Planare Graphen mit Maximalgrad 3	stark NP-schwer [Lehmann et al., 2014]	✗
Beliebige Graphen	nicht-APX [Lehmann et al., 2014]	✗

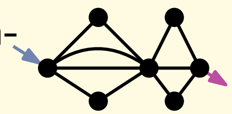
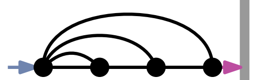
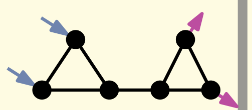
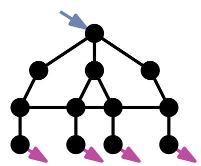
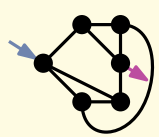
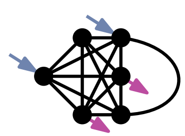
ein Erzeuger, ein Verbraucher

$|V_G| = 2$, $|V_D| = 2$

Komplexität

- Was passiert, wenn wir die Anzahl der switches minimieren oder eine Menge von nicht-schaltbaren Kanten fixieren?
- Existiert ein PTAS auf Kakteen für MTSF?
- Ersetze ✗ durch ✓

Offene Probleme

Problem	Graphenstruktur	$\lvert V_G \rvert$	$\lvert V_D \rvert$	b	cap	Komplexität	Algorithmen
MTSF und OTS	Penrose-Minoren-freie Graphen 	1	1	—	—	Polynomialzeit lösbar [Grastien et al., 2018]	✓ besser? ∞ ∞
MTSF und OTS	Serienparallele Graphen 	1	1	∞ ∞	∞ 1	[Kocuk et al., 2016] NP-schwer [Grastien et al., 2018]	✗
MTSF und OTS	Kakteen mit Maximalgrad 3 	∞	∞	1	∞	NP-schwer [Lehmann et al., 2014]	✓ besser? ∞ ∞
MTSF und OTS	2-Level-Bäume 	1	∞	∞	∞	NP-schwer [Lehmann et al., 2014]	✗
MTSF und OTS	Planare Graphen mit Maximalgrad 3 	1	1	∞	1	stark NP-schwer [Lehmann et al., 2014]	✗
MTSF OTS	Beliebige Graphen 	2 1	2 ∞	∞	∞	nicht-APX [Lehmann et al., 2014]	✗

Andere interessante Strukturen?

Stärkere Ergebnisse?

Irgendein Algorithmus, der Garantien gibt?

Referenzen

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. Laurence A. Wolsey *Integer Programming.* Band 52 von Wiley Series in Discrete Mathematics and Optimization, John Wiley & Sons. ISBN: 9780471283669, 1998.
3. Jack Edmonds and Richard M. Karp *Theoretical Improvements in Algorithmic Efficiency for Network Flow Problems.* Journal ACM, 19(2):248–264. DOI: 10.1145/321694.321699, 1972.
4. R. K. Ahuja and James B. Orlin *A Fast and Simple Algorithm for the Maximum Flow Problem.* Journal on Operations Research, 37(5):748–759. DOI: 10.1287/opre.37.5.748, 1989.
5. Hassler Whitney *On the Abstract Properties of Linear Dependence.* American Journal of Mathematics, 57(3):509–533. DOI: 10.2307/2371182, 1935.
6. Gustav Robert Kirchhoff *Über die Auflösung der Gleichungen, auf welche man bei der Untersuchung der linearen Vertheilung galvanischer Ströme geführt wird.* Annalen der Physik, 148(12):497–508. DOI: 10.1002/andp.18471481202, 1847.
7. J. D. Horton *A Polynomial-Time Algorithm to Find the Shortest Cycle Basis of a Graph* SIAM Journal on Computing, 16(2):358–366. DOI: 10.1137/0216026, 1987.
8. Sundaram Seshu and Myril B. Reed *Linear Graphs and Electrical Networks* Addison-Wesley Publishing Company, 1961.