Algorithmentechnik — Vorlesung am 13. Dez. 2005

http://i11www.ira.uka.de/teaching/WS_0506/algotech

Steffen Mecke (mecke@ira.uka.de)

WS 05/06

Flussprobleme und Dualität

- Grundlagen
- Das Max-Flow Min-Cut Theorem
- Ford/Fulkerson
- Goldberg/Tarjan

Definition

Gegeben sei ein einfacher, gerichteter Graph $D = (V, E)$ mit *Kantenkapazitäten* $c\colon E \to \mathbb{R}_0^+$ und ausgezeichneten Knoten $s, t \in V$, s *Quelle*, t *Senke*.
$(D; s; t; c)$ heißt *Netzwerk*.
Eine Abbildung $f\colon E \to \mathbb{R}_0^+$ heißt *Fluss*, wenn

(i) Für alle $(i, j) \in E$ gilt

$$0 \leq f(i,j) \leq c(i,j) \qquad (\textit{Kapazitätsbedingung})$$

(ii) Für alle $i \in V \setminus \{s, t\}$ gilt

$$\sum_{\{j \mid (i,j) \in E\}} f(i,j) - \sum_{\{j \mid (j,i) \in E\}} f(j,i) = 0$$

$(\textit{Flusserhaltungsbedingung})$

Definition II

Definition

Der Ausdruck

$$w(f) := \sum_{(s,i)\in E} f(s,i) - \sum_{(i,s)\in E} f(i,s)$$

heißt *Wert* des Flusses f.

Problem

In einem Netzwerk $(D; s, t; c)$ soll ein Maximalfluss gefunden werden, d.h. ein Fluss mit maximalem Wert.

Definition

Definition

- Eine Menge $S \subset V$ induziert eine Partition $(S, V \setminus S)$ der Knotenmenge V, die wir *Schnitt* nennen.
- $(S, V \setminus S)$ heißt ein s-t-Schnitt, wenn $s \in S$ und $t \in V \setminus S$.
- Die *Kapazität* eines Schnittes $(S, V \setminus S)$ ist definiert als

$$c(S, V \setminus S) := \sum_{\substack{(i,j) \in E \\ i \in S \\ j \in V \setminus S}} c(i,j) \ .$$

- Ein Schnitt mit minimaler Kapazität heißt *minimaler Schnitt*.

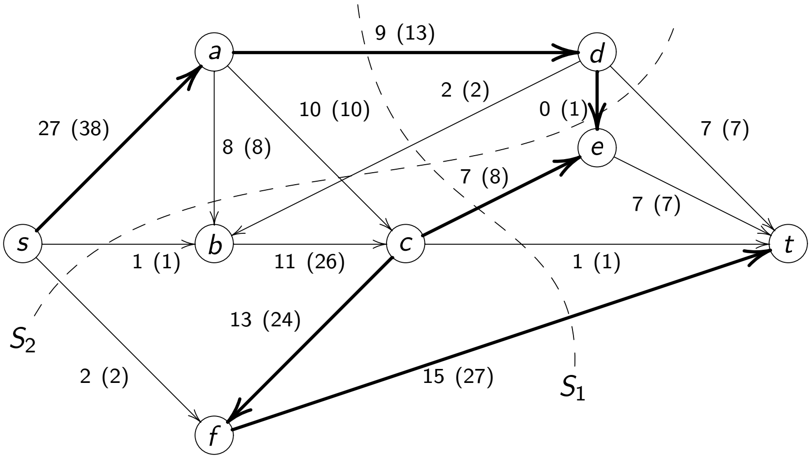
Lemma (Schnittlemma)

Sei $(S, V \setminus S)$ ein s-t-Schnitt im Netzwerk $(D; s, t; c)$. Für jeden Fluss f gilt, dass

$$w(f) = \sum_{\substack{(i,j) \in E \\ i \in S \\ j \in V \setminus S}} f(i,j) - \sum_{\substack{(i,j) \in E \\ j \in S \\ i \in V \setminus S}} f(i,j) \ .$$

Insbesondere ist $w(f) \leq c(S, V \setminus S)$.

Beispiel


a
d
9 (13)
2 (2)
10 (10)
0 (1)
27 (38)
8 (8)
e
7 (7)
7 (8)
7 (7)
s
b
c
t
1 (1)
11 (26)
1 (1)
13 (24)
S2
2 (2)
15 (27)
S1
f

Erhöhender Weg

Definition

Zu einem Fluß f im Netzwerk $(D; s, t; c)$ betrachten wir einen (ungerichteten) Weg von s nach t.

- Alle Kanten auf diesem Weg, die von s in Richtung t gerichtet sind, heißen *Vorwärtskanten*,
- alle anderen *Rückwärtskanten*.
- Ein solcher Weg heißt *erhöhender Weg* (bezüglich f), wenn

$$\begin{array}{ll} f(i,j) < c(i,j) & \text{für jede Vorwärtskante } (i,j) \\ f(i,j) > 0 & \text{für jede Rückwärtskante } (i,j) \end{array}$$

Satz vom erhöhenden Weg

Satz

Ein Fluss f in einem Netzwerk $(D; s, t; c)$ ist genau dann ein Maximalfluss, wenn es bezüglich f keinen erhöhenden Weg gibt.

Beweisskizze.

„⇒“: Erhöhung des Flusses entlang eines erhöhenden Weges.
„⇒“: Konstruktion eines „saturierten“ Schnittes und Schnittlemma. □

Max-Flow Min-Cut Theorem

Satz (Ford/Fulkerson, sowie Elias/Feinstein/Shanon ’56)

In einem Netzwerk $(D; s, t; c)$ ist der Wert eines Maximalflusses gleich der minimalen Kapazität eines s-t-Schnittes.

Beweis.

- Sei f ein Maximalfluss.
- Sei S die Menge aller Knoten, die auf einem erhöhenden Weg von s aus erreichbar sind.
- $(S, V \setminus S)$ ist ein s-t-Schnitt und
- $w(f) = c(S, V \setminus S)$
- $c(S, V \setminus S) = \min\limits_{\substack{\emptyset \neq S' \neq V \\ s \in S' \\ t \in V \setminus S'}} c(S', V \setminus S')$

$\square$

Folgerung

Für einen Fluß f in einem Netzwerk $(D; s, t; c)$ ist äquivalent:

(i) Der Wert $w(f)$ ist maximal.

(ii) Es gibt keinen bezüglich f erhöhenden Weg.

(iii) Die Kapazität eines minimalen s-t-Schnittes $(S, V \setminus S)$ ist $w(f)$.

Ganzzahligkeitssatz

Satz

Sei $(D; s, t; c)$ ein Netzwerk mit $c\colon E \longrightarrow \mathbb{N}_0$. Dann gibt es einen Maximalfluss f mit $f(i,j) \in \mathbb{N}_0$ für alle $(i,j) \in E$ und damit $w(f) \in \mathbb{N}_0$.

Beweis.

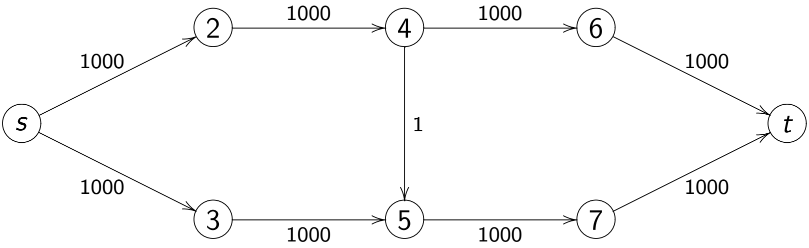
- Definieren ganzzahligen „Anfangsfluss“ $f_0\colon E \longrightarrow \mathbb{N}_0$ (zum Beispiel $f_0(i,j) = 0$ für alle $(i,j) \in E$).
- Ist f_0 nicht maximal, so existiert ein erhöhender Weg bezüglich f_0. Für diesen ist das $\Delta_0 > 0$ ganzzahlig.
- Erhöhe f_0 um Δ: $w(f_1) = w(f_0) + \Delta_0$, f_1 ist wiederum ganzzahlig.
- Iteriere, bis ein ganzzahliger Fluss f_i erreicht ist, bezüglich dessen es keinen erhöhenden Weg mehr gibt.

Ford-Fulkerson-Algorithmus

1. Setze $f(i,j) := 0$ für alle Kanten $(i,j) \in E$.
2. Solange es einen erhöhenden Weg bezüglich f gibt, führe aus:
   2.1 Sei $\langle e_1, e_2, \ldots, e_k \rangle$ mit $e_1, \ldots, e_k \in E$ erhöhender Weg.
   2.2 Setze $\Delta := \min(\{c(e_i) - f(e_i) | \ e_i \text{ VwK}\} \cup \{f(e_i) | \ e_i \text{ RwK}\})$
   2.3 Setze $f(e_i) := f(e_i) + \Delta$, falls e_i eine Vorwärtskante ist, und setze $f(e_i) := f(e_i) - \Delta$, falls e_i eine Rückwärtskante ist.

Laufzeit: Bei nichtrationalen Werten $c(i,j)$ kann es passieren, dass das Verfahren nicht terminiert. Bei rationalen Werten geht $C := \max\{c(i,j) | \ (i,j) \in E\}$ im Allgemeinen in die Laufzeit ein.

Böses Beispiel

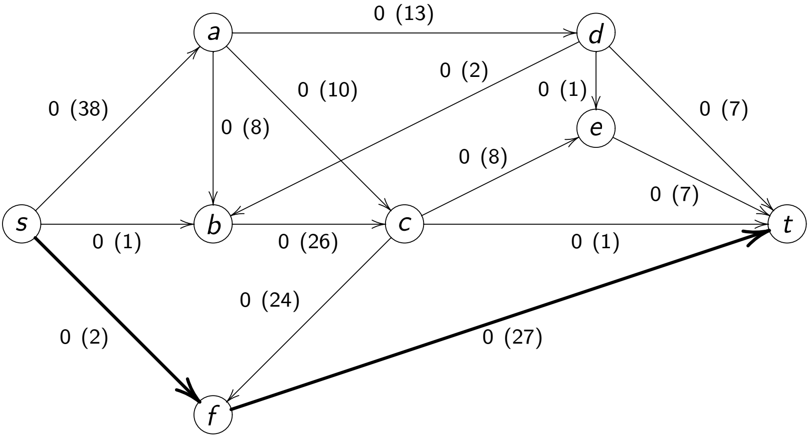


Der Algorithmus von Edmonds und Karp (1972)

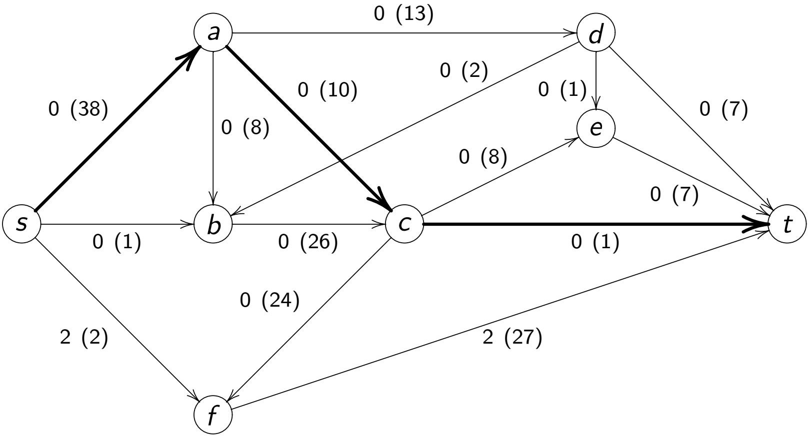
Findet einen erhöhenden Weg kann man systematisch mittels Breitensuche. Man kann zeigen:

- Der Fluss wird maximal $O(|V||E|)$ oft erhöht.
- Jede Erhöhung kostet $O(|E|)$.
- Gesamtaufwand $O(|V||E|^2)$.

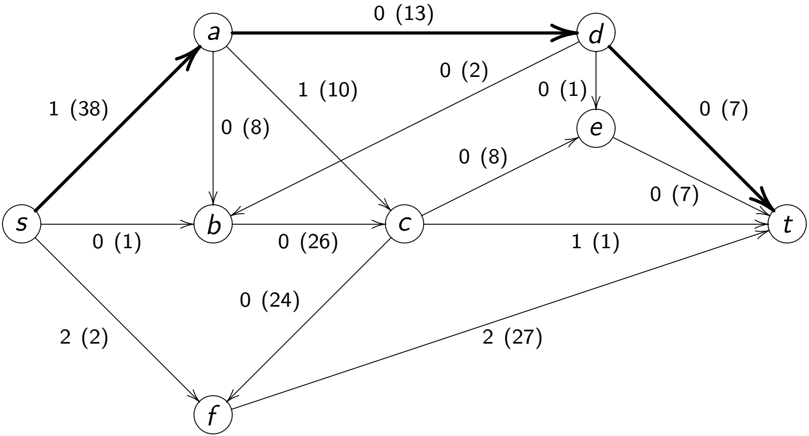
Beispiel



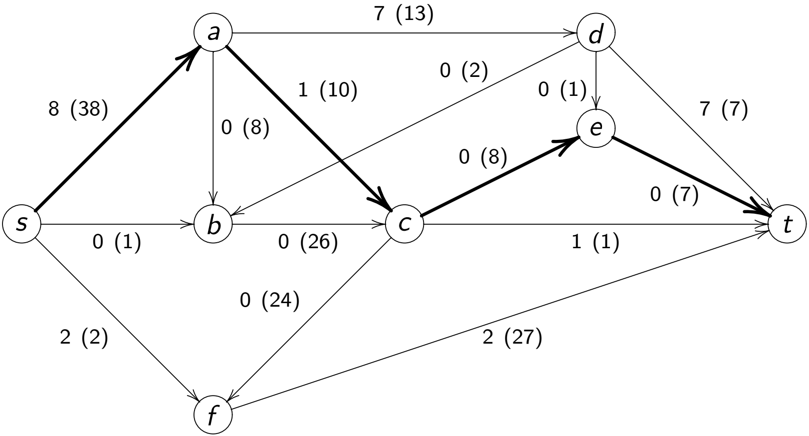
Beispiel



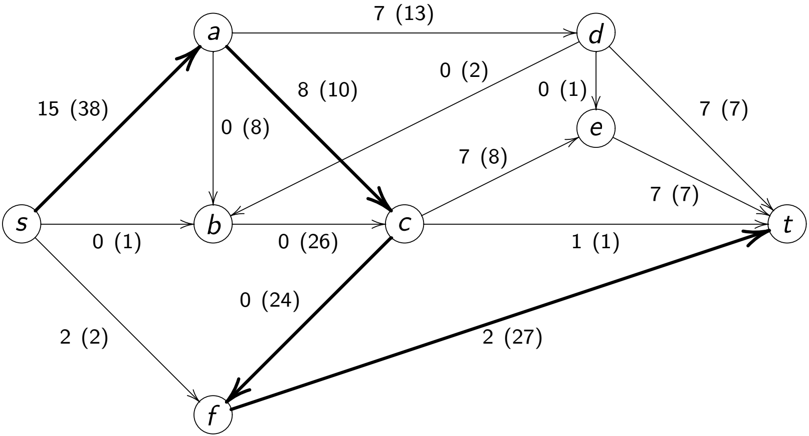
Beispiel



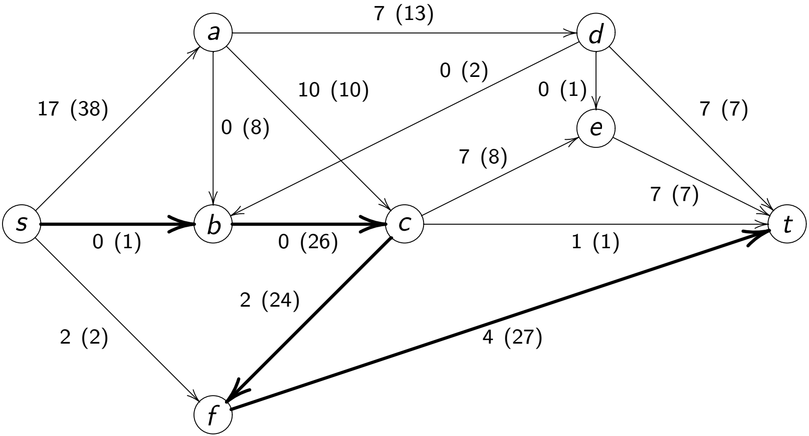
Beispiel



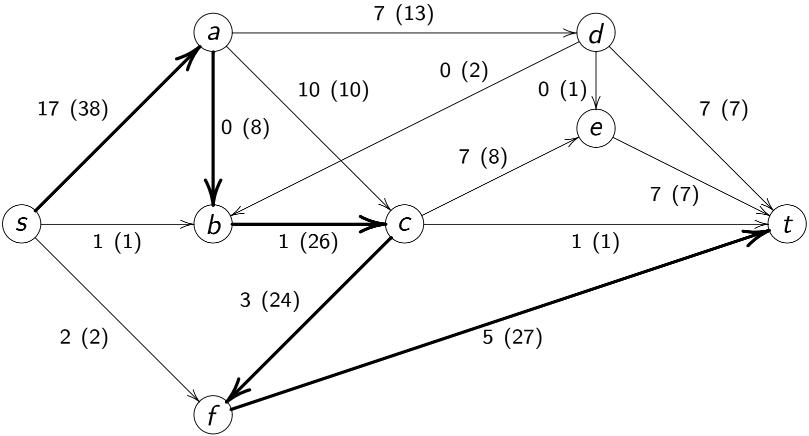
Beispiel



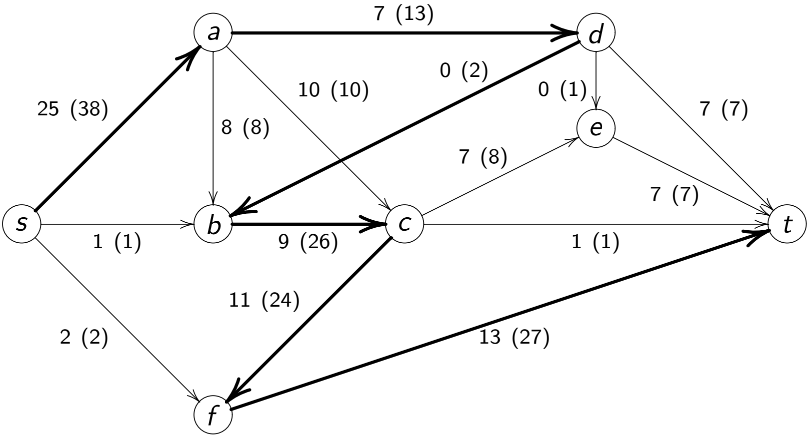
Beispiel



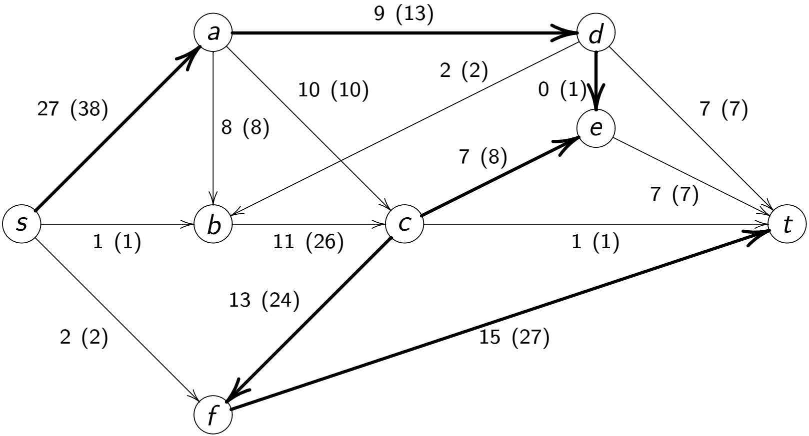
Beispiel



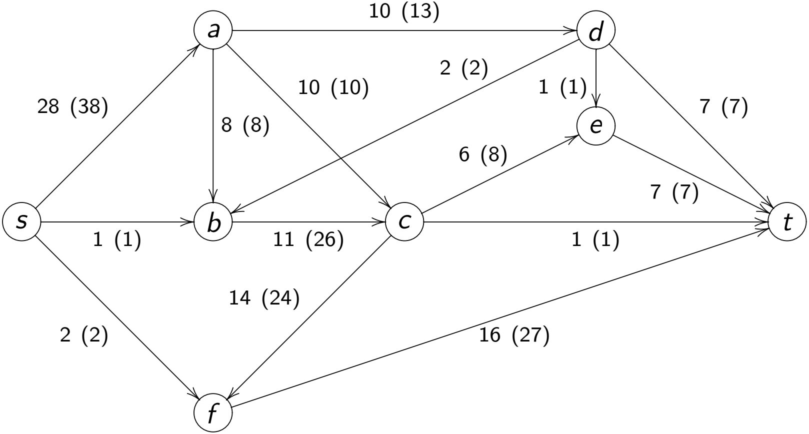
Beispiel



Beispiel



Beispiel


a
d
s
b
c
e
t
f
10 (13)
2 (2)
10 (10)
1 (1)
28 (38)
8 (8)
7 (7)
6 (8)
7 (7)
1 (1)
11 (26)
1 (1)
14 (24)
2 (2)
16 (27)

Der Algorithmus von Goldberg und Tarjan (1988)

- Effizientester bekannter Algorithmus für Max Flow.
- Laufzeit $O(|V|^2|E|)$.
- Beruht auf der Verwendung von Präflüssen.
- Grundidee: Aus Knoten mit „Flussüberschuss“ wird Überfluss in Richtung t geschoben (PUSH)...
- ...auf „ungefähr“ kürzesten Wegen (RELABEL).

Äquivalente Formulierung

- Erweitere $D = (V, E)$ zu $D' = (V, V' \times V')$ und
- $c\colon E \longrightarrow \mathbb{R}_0^+$ zu $c'\colon V \times V \longrightarrow \mathbb{R}_0^+$, durch

$$c'(v, w) = 0 \text{ für } (v, w) \notin E \quad .$$

- Ein Fluss ist dann eine Abbildung $f\colon V \times V \longrightarrow \mathbb{R}$ mit
  - *Kapazitätsbedingung*

$$\forall (v, w) \in V \times V \quad f(v, w) \leq c'(v, w) \quad , \tag{1}$$

  - *Antisymmetrie-Forderung*

$$\forall (v, w) \in V \times V \quad f(v, w) = -f(w, v) \quad \text{und} \tag{2}$$

  - *Flußerhaltungsbedingung*

$$\forall v \in V \setminus \{s, t\} \quad \sum_{u \in V} f(u, v) = 0 \quad . \tag{3}$$

Äquivalente Formulierung (Forts.)

Der *Wert* eines Flusses f ist dann

$$w(f) = \sum_{v \in V} f(s, v) = \sum_{v \in V} f(v, t) \ .$$

Die Antisymmetriebedingung (2) bewirkt, dass nicht beide Kanten (v, w) und (w, v) „echten“ Fluss tragen. Dadurch wird die Flusserhaltungsbedingung und die Berechnung des Flusswertes vereinfacht.

Präfluss

Definition

Ein *Präfluss* ist eine Abbildung $f\colon V \times V \to \mathbb{R}$, welche die Kapazitätsbedingung und die Antisymmetrie-Forderung erfüllt sowie

$$\forall v \in V \setminus \{s\} \quad \sum_{u \in V} f(u,v) \geq 0 \ . \tag{4}$$

Die Bedingung besagt, dass für alle Knoten $v \in V \setminus \{s\}$ mindestens soviel Fluss hineinfließt wie auch hinausfließt.

Definition

Sei f ein Präfluss. Für $v \in V$ heißt der Wert

$$e(v) := \sum_{u \in V} f(u,v)$$

Flussüberschuss, und die Abbildung $r_f \colon E' \to \mathbb{R}$ mit

$$\forall (u,v) \in E' \quad r_f(u,v) := c'(u,v) - f(u,v)$$

heißt *Restkapazität*.

Definition

Eine Kante $(v, w) \in E'$ heißt *Residualkante* bezüglich Präfluss f, falls $r_f(v, w) > 0$. Der *Residualgraph* zu f ist gegeben durch $D_f(V, E_f)$ mit

$$E_f := \{(v, w) \in E' \mid r_f(v, w) > 0\}.$$

Definition

Eine Abbildung $\operatorname{dist}\colon V \to \mathbb{N}_0 \cup \{\infty\}$ heißt *zulässige Markierung* bezüglich eines Präflusses f, falls

$$\operatorname{dist}(s) = |V|, \operatorname{dist}(t) = 0$$

und für alle $v \in V \setminus \{s, t\}$, falls $(v, w) \in E_f$,

$$\operatorname{dist}(v) \leq \operatorname{dist}(w) + 1$$

gilt. Ein Knoten $v \in V$ heißt *aktiv* im Laufe des Algorithmus, wenn

$$v \in V \setminus \{s, t\}, e(v) > 0 \text{ und } \operatorname{dist}(v) < \infty.$$