Flüsse in Netzwerken

Seminar über Algorithmen

SoSe 2005

Mike Rohland & Julia Schenk

Inhalt

- Einführung
- Definition
- Maximale Flüsse
- Schnitte
- Restgraphen
- Zunehmende Wege
- Max-Fluss Min-Schnitt Theorem
- Ford-Fulkerson Methode
- Edmonds-Karp Algorithmus
- Maximale Bipartite Matchings

Problemstellungen

- Welchen Verkehrsfluss (in Fahrzeugen pro Minute) kann ich höchstens durch eine Stadt leiten, deren Straßennetz gegeben ist?

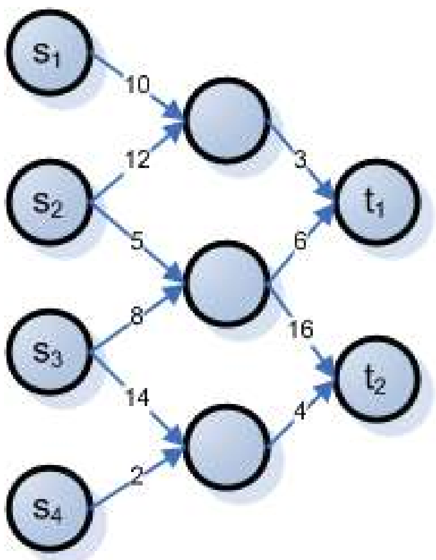
- Welche Wassermenge kann ich durch die Kanalisation höchstens abtransportieren?

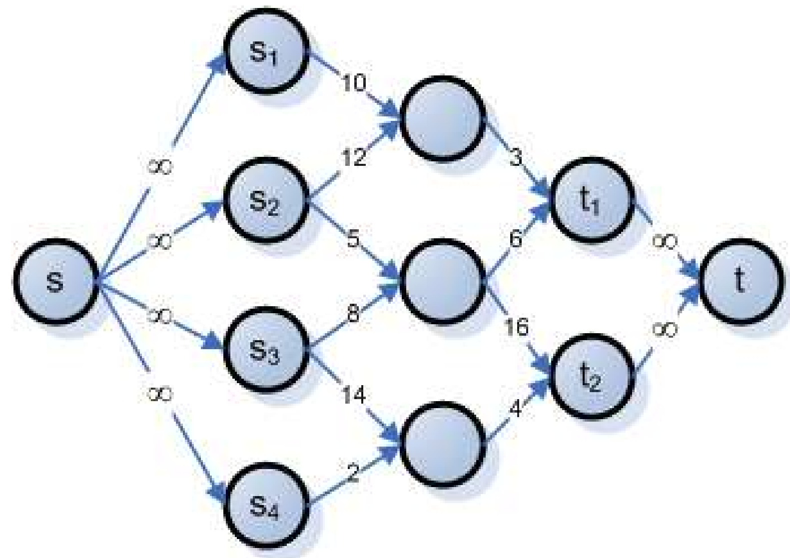
➔ Wir betrachten dabei das Problem der maximalen Flüsse!

Graphentheoretische Definition

- G = (V, E) gerichteter Graph G mit
- c: E → R^+ Kapazitätsfunktion
- Quelle s, Senke t
- Fluss f: V x V → **R**
  - ***Kapazitätsbeschränkung*: Für alle** $u, v \in V$ **gilt:** $f(u,v) \le c(u,v)$
  - **Schiefe Symmetrie: Für alle** $u, v \in V$ **gilt** $f(u,v) = -f(v,u)$
  - ***Flusserhaltung*: Für alle** $u \in V - \{s,t\}$ **ist** $\sum_{v \in V} f(u,v) = 0$
  - **Größe f(u,v) ist der Fluss von Knoten u zu Knoten v.**
  - **Wert eines Flusses:** $|f| = \sum_{v \in V} f(s,v)$
  - **Nettofluss in einem Knoten ist die Summe aller den Knoten verlassenden Flüsse minus aller in den Knoten hinein fließenden Flüsse.**

Fluss mit mehreren Quellen / Senken





- hinzufügen einer Superquelle s und einer Supersenke t

Maximale Flüsse

- Ein maximaler Fluss f in G ist ein Fluss f mit maximalem Wert f unter allen Flüssen in G.

- Wie groß kann ein maximaler Fluss überhaupt sein ?

  – In einem Netzwerk kann nicht mehr fließen, als aus der Quelle hinaus fließt oder in die Senke hineinfließt.
  – Außerdem begrenzt jeder *Schnitt* durch den Graphen, der s von t trennt den Wert eines max. Flusses

Schnitte in Flussnetzwerken I

- Zerlegung der Knotenmenge V in S und T
  - mit $s \in S$ und $t \in T$
- Der **Nettofluss** in einem Schnitt ist die Summe des positiven Flusses von S nach T minus dem von T nach S
- Die **Kapazität** c(S,T) eines Schnittes S,T ist die Summe der Kapazitäten aller Kanten die von S nach T führen
- Ein Schnitt mit kleinster Kapazität unter allen möglichen Schnitten heißt **minimaler Schnitt**.
- Der Nettofluss ist über jeden Schnitt gleich.

Schnitte in Flussnetzwerken II

- **Lemma:**

  Sei f ein Fluss in einem Flussnetzwerk G=(V, E), mit Quelle s und Senke t, und sei (S, T) ein Schnitt von G. Dann ist der Nettofluss über (S, T) $f(S, T) = | f |$

- **Korollar:**

  Der Wert eines jeden Flusses f in einem Flussnetzwerk G wird durch die Kapazität eines beliebigen Schnittes (S,T) begrenzt.

Restgraphen

- Der Restgraph G_f zu einem Fluss f beschreibt gerade alle Flussvergrößerungsmöglichkeiten
- Restgraph $G_f(V,E_f)$ mit $E_f = \{(u,v) \in V \times V : c_f(u,v) > 0\}$
- Restkapazität $c_f(u,v) = c(u,v) – f(u,v)$
- Der Restgraph besteht aus Kanten, deren Kapazitäten nicht voll ausgeschöpft sind.
- Vorwärtskante: $f(u,v) < c(u,v)$ (Restkapazität vorhanden)
- Rückwärtskante: $c_f(v,u) = c(u,v) – f(u,v) > 0$
- Jeder Weg von S nach T im Restgraphen ist ein zunehmender Weg
- Den gesamten Fluss kann man höchstens um das Minimum der gefundenen Restkapazitäten erhöhen.

Zunehmende Wege

- Jeder Weg von s nach t im Restgraphen ist ein zunehmender Weg.
- Ein Fluss ist genau dann maximal, wenn es für f keinen zunehmenden Weg gibt.
  - Denn wenn es einen zunehmenden Weg für einen Fluss f gibt, dann könnte man den Fluss entlang dieses Weges vergrößern.

Max. Fluss – min. Schnitt Theorem I

- Sei f ein Fluss in einem Flussnetzwerk G = (V,E) , mit Quelle s und Senke t, dann gelten folgende Bedingungen:
  (1) f ist ein maximaler Fluss in G
  (2) Der Restgraph G_f enthält keine zunehmenden Wege
  (3) Der Wert des Flusses f entspricht der Kapazität eines Schnittes: $| f | = c(S,T)$ für einen Schnitt (S,T) in G

Beweis (1) → (2):

Sofern es für einen maximalen Fluss einen zunehmenden Weg gibt, kann dieser Fluss nicht maximal sein

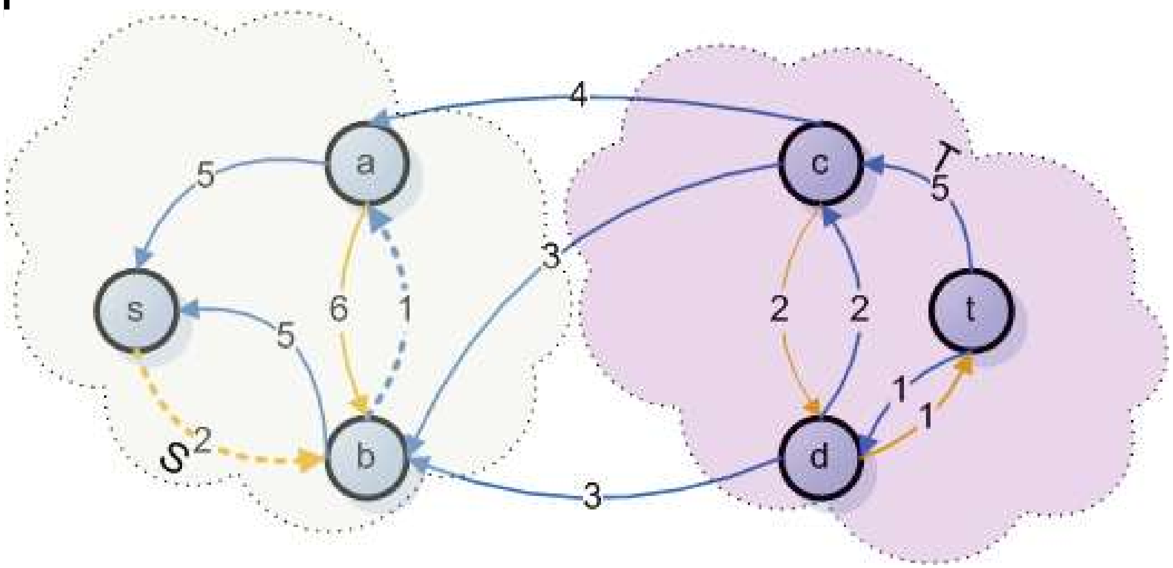
Max. Fluss – min. Schnitt Theorem II

Beweis (2) → (3)

- $S = \{v \in V$, es existiert ein Weg von s nach v in $G_f\}$
- $T = V - S$
- (S,T) ist Schnitt
- | f | = c (S, T)

Beweis(3) → (1)

- der Schnitt ist minimal
- daher muss f maximal sein



Ford-Fulkerson Methode

- Methode zur Lösung des Problems des Maximalen Flusses
- Benutzt dazu
  - Restgraphen
  - Zunehmende (augmentierende) Wege
- iterativ

Ford-Fulkerson Methode - Pseudocode

```
for each edge (u,v) ϵ E
      do f[u,v] ← 0
           f[v,u] ← 0
while es gibt einen Weg p von s nach t im Restgraphen
      do cf(p) ← min {cf(u,v): (u,v) ist auf p}
            for jede Kante (u,v) in p
                   do f[u,v] ← f[u,v] + cf(p)
                        f[v,u] ← - f[v,u]
```

Die Laufzeit des Algorithmus hängt maßgeblich davon ab wie die zunehmenden Wege gefunden werden

Ford-Fulkerson Methode – Beispiel I

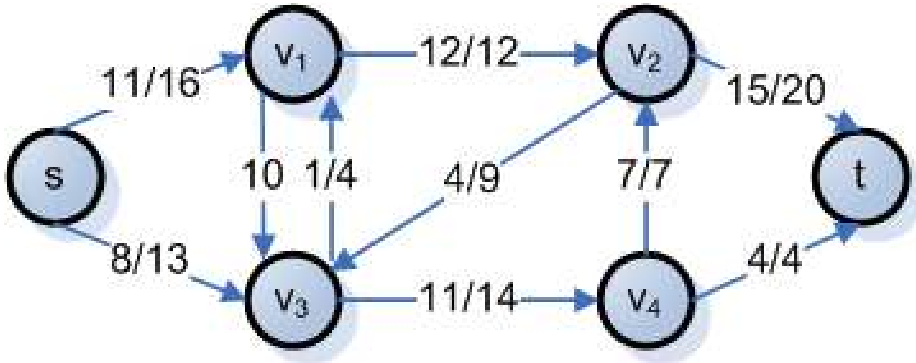
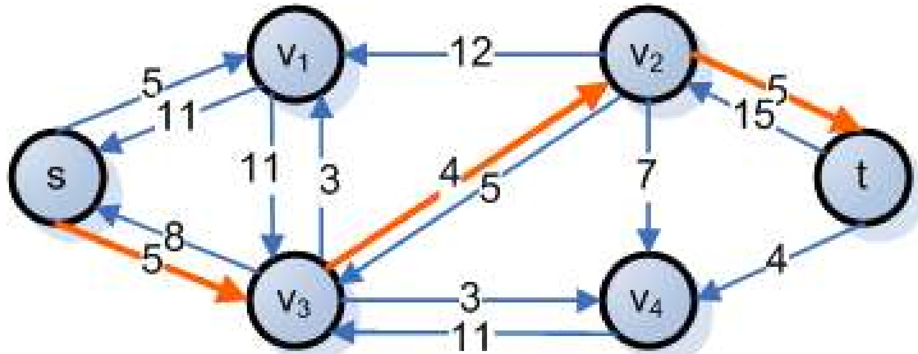


Abbildung 1
Flussnetzwerk

Fluss / Kapazität

Abbildung 2
Restgraph mit zunehmendem Weg

$C_f(p) = c(v_3,v_2) = 4$



Ford-Fulkerson Methode – Beispiel II

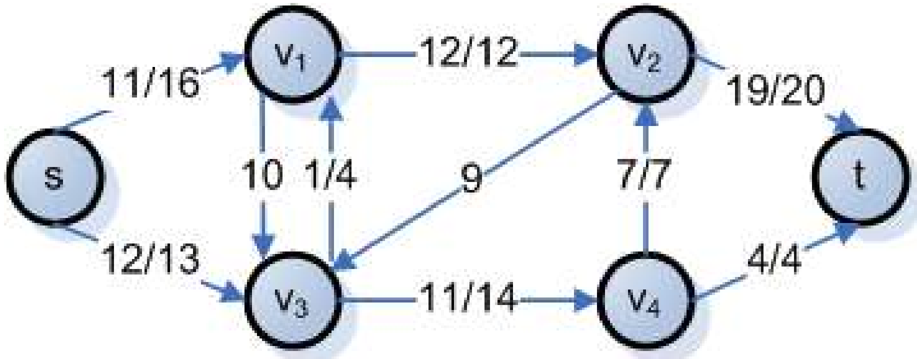
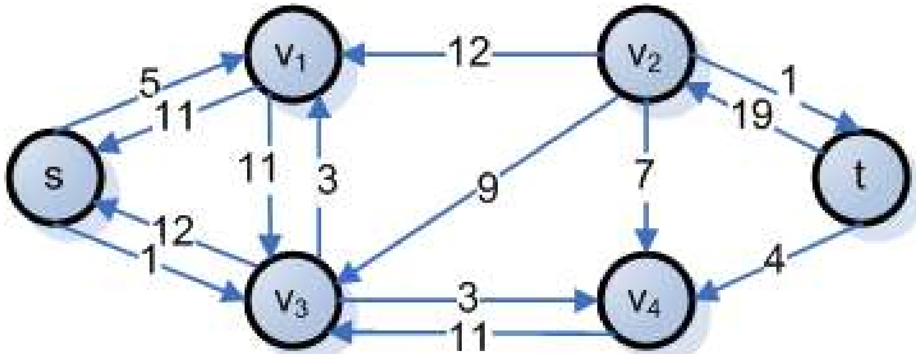


Abbildung 3
Flussnetzwerk mit zunehmendem Weg aus 3

Abbildung 4
Restgraph des maximalen Flusses



Ford-Fulkerson Methode - Analyse

```
for each edge (u,v) ϵ E
      do f[u,v] ← 0
         f[v,u] ← 0                                              Θ(E)
while es gibt einen Weg p von s nach t im Restgraphen
      do c_f(p) ← min {c_f(u,v): (u,v) ist auf p}               O(E)
              for jede Kante (u,v) in p
                   do f[u,v] ← f[u,v] + c_f(p)
                      f[v,u] ← - f[v,u]                          O(E)
                                                             f* - Mal
```

$O(E\ |f^*|\ )$

Edmonds-Karp Algorithmus

- Verbesserte Variante der Ford-Fulkerson Methode
- Hierbei wird der zunehmende Weg durch Breitensuche gefunden
- Daher muss der jeweils gefundene zunehmende Weg ein kürzester Weg im Restgraphen von s nach t sein
- In jedem Knoten wird zusätzlich die Entfernung zur Quelle s gespeichert
- Läuft in $O(V E^2)$

Edmonds-Karp Alg. - Lemma

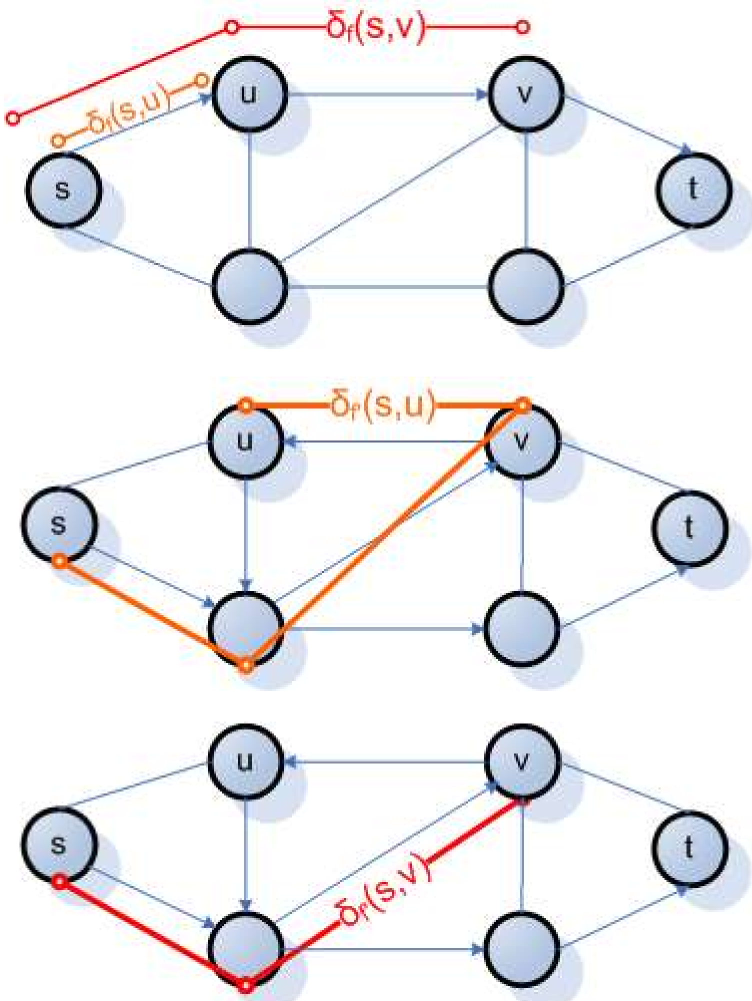
- Sei G = (V,E) ein Flussnetzwerk mit Quelle s und Senke t.
- Dann gilt beim Edmonds-Karp Algorithmus für alle Knoten $v \in V - \{s,t\}$ das die Entfernung $\delta_f(s,v)$ (von der Quelle s zu einem Knoten v) im Restgraphen G_f mit jeder Flusserhöhung monoton wächst.

Edmonds-Karp Alg. – Theorem I

- Wenn der Edmonds-Karp Algorithmus auf einem Flussnetzwerk G = (V,E) mit Quelle s und Senke t läuft, dann ist die Anzahl der durch den Algorithmus durchgeführten Flusserhöhungen O(V E).

- Zu zeigen: jede der |E| Kanten kann höchstens |V| / 2 -1 mal kritisch werden

Edmonds-Karp Alg. – Theorem II



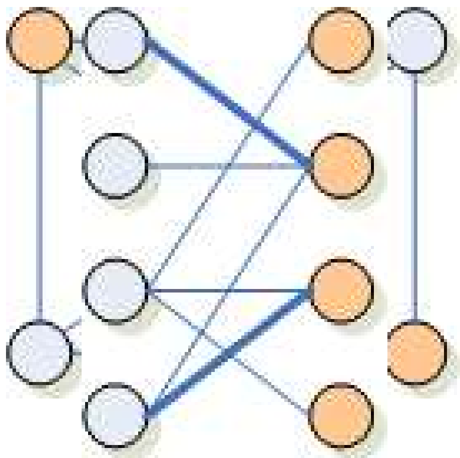
$\delta_f(s,v) = \delta_f(s,u) + 1$

$\delta_{f'}(s,u) = \delta_{f'}(s,v) + 1$

$$
\begin{aligned}
\delta_f(s,v) &\leq \delta_{f'}(s,v) \\
\delta_{f'}(s,u) &= \delta_{f'}(s,v) + 1 \\
&\geq \delta_f(s,v) + 1 \\
&\geq \delta_f(s,u) + 2
\end{aligned}
$$

Maximale Bipartite Matchings I

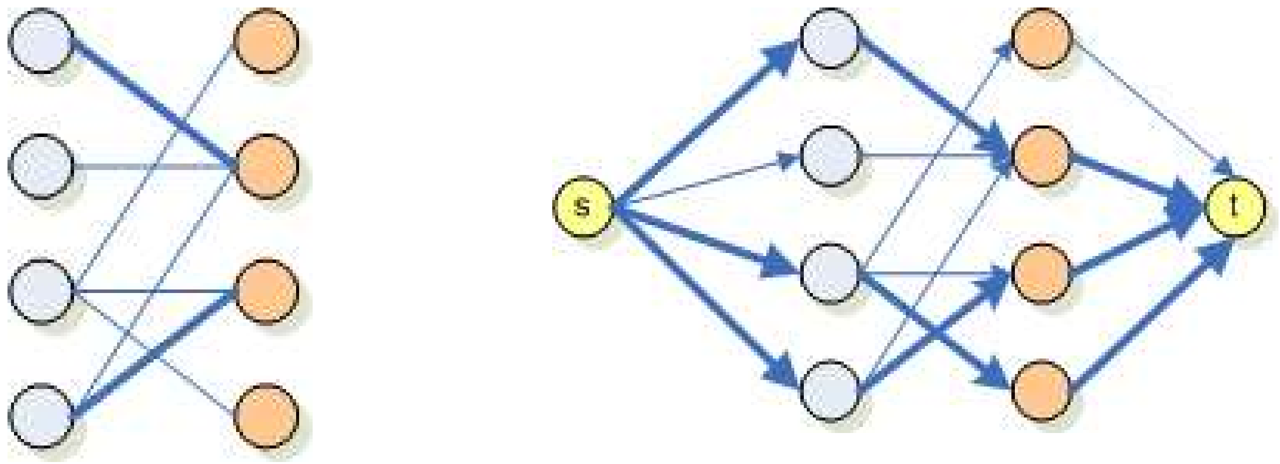
- **Bipartiter Graph**
- G = (V, E)
- V = L U R, L ∩ R = ø und
  Kante (l, r) mit l ϵ L und r ϵ R



- **Bipartites Matching**
- M ~~E, für alle v ϵ~~ V existiert
  maximal eine Kante in M
- **Maximales Matching**
  |M| ≥ |M‘| für alle M‘

- **Problem**
  finde in einem Bipartitem Graph ein maximales Matching

Maximale Bipartite Matchings II

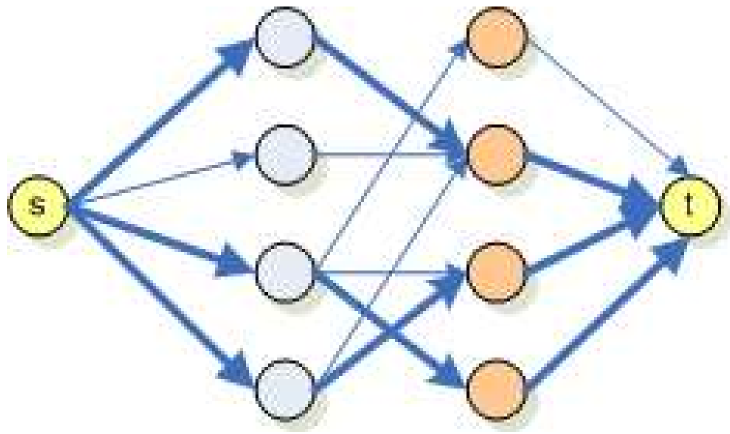


Umwandlung:

- Hinzufügen von s und t
- Ausrichten der vorhandenen Kanten von L nach R
- Neue Kanten von s nach L und R nach t hinzufügen
- Jeder Kante gleich bleibende Kapazität (1) zuweisen

Finden des Maximalen Matchings I

- Das dem bipartien Graphen G entsprechende Flussnetzwerk G‘ = (V‘, E‘) ist wie folgt definiert:
  - Quelle s und Senke t sind neue Knoten nicht in V und V‘ = V U {s,t}
  - Wenn die Aufteilung der Knoten von G gleich V = L U R ist, dann sind die gerichteten Kanten von G‘ die Kanten von E, gerichtet von L nach R, mit V neuen Kanten
  - Letztendlich ordnet man jeder Kante von E‘ eine Kapazität von eins zu

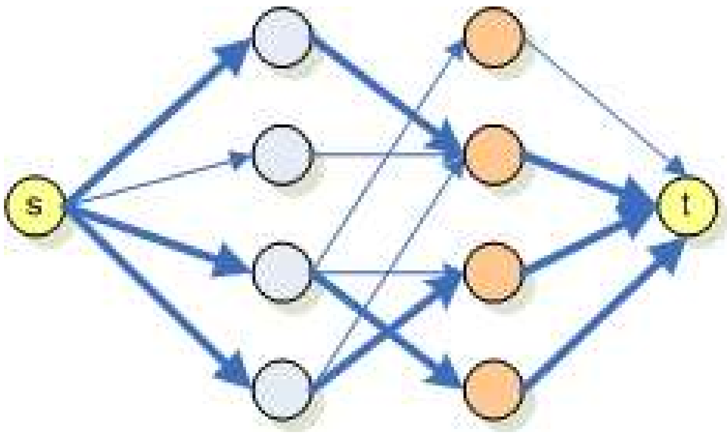


Finden des Maximalen Matchings II

Lemma

Sei G = (V,E) ein bipartiter Graph mit der Knotenaufteilung V= L U R und sei G‘ = (V‘, E‘) das G entsprechende Flussnetzwerk.

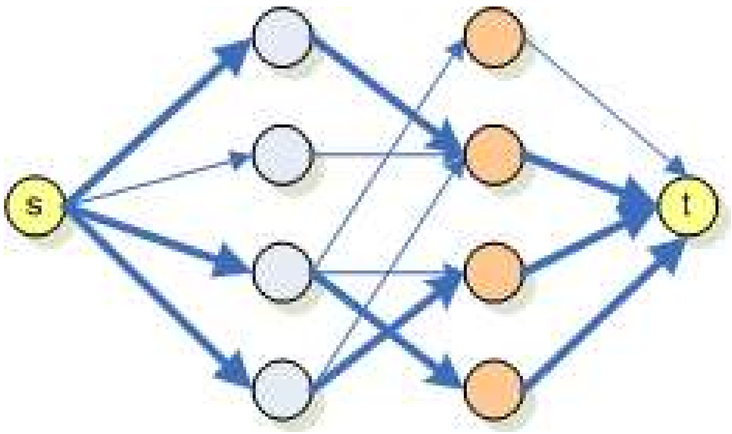
Wenn M ein Matching in G ist, dann gibt es einen Fluss f in G‘ mit dem Flusswert |f| = |M|. Umgekehrt, wenn f ein Fluss in G‘ ist, dann gibt es ein Matching M in G mit der Kardinalität |M| = |f|.



Finden des Maximalen Matchings III

Korollar

Die Kardinalität eines maximalen Matchings M in einem bipartitem Graph G ist der Wert eines maximalen Flusses f in seinem entsprechendem Flussnetzwerk G‘.



Quellen

- **Introduction to Algorithms**
  - Cormen, Leiserson, Rivest, Stein
  - 2nd Edition
- **Algorithmen und Datenstrukturen**
  - Ottmann, Widmayer
  - 3. Auflage