Variante der Ford–Fulkerson–Methode

Dieser Algorithmus funktioniert bei ganzzahligen Kapazitäten:

Algorithmus

$K \leftarrow 2^{\lfloor \log_2(\max\{ c(u,v) \mid (u,v) \in E \}) \rfloor}$
$f \leftarrow 0$
while $K \geq 1$ **do**
  while es gibt einen augmentierenden
    Pfad p mit $c_f(p) \geq K$ **do**
     augmentiere f entlang p
  $K \leftarrow K/2$
return f

Die Laufzeit ist $O(|E|^2 \log K)$.
$\Rightarrow$ vergleiche mit $O(|E| f^*)$.

Variante der Ford–Fulkerson–Methode

Theorem

Die Laufzeit dieser Variante beträgt $O(|E|^2 \log C)$*, wobei* $C = \max\{\, c(u,v) \mid (u,v) \in E \,\}$.

Beweis.

- Die Restkapazität eines minimalen Schnitts ist stets höchstens $2K|E|$.
- Für jedes K gibt es nur $|E|$ Augmentierungen
- Es gibt $O(\log C)$ verschiedene K

$\square$

Der Edmonds–Karp–Algorithmus

Die Ford–Fulkerson–Methode kann sehr langsam sein, auch wenn das Netzwerk klein ist.

Der Edmonds–Karp–Algorithmus ist polynomiell in der Größe des Netzwerks.

Algorithmus

Initialisiere Fluß f zu 0
while es gibt einen augmentierenden Pfad **do**
  finde einen kürzesten augmentierenden Pfad p
  augmentiere f entlang p
return f

Unterschied: Es wird ein kürzester Pfad gewählt

Der Edmonds–Karp–Algorithmus

Algorithmus

for each edge $(u, v) \in E$ **do**
  $f(u, v) \leftarrow 0$
  $f(v, u) \leftarrow 0$
while there exists a path from s to t in G_f **do**
  $p \leftarrow$ a shortest path from s to t in G_f
  $c_f(p) \leftarrow \min\{\, c_f(u, v) \mid (u, v)$ is in $p\,\}$
  for each edge (u, v) in p **do**
    $f(u, v) \leftarrow f(u, v) + c_f(p)$
    $f(v, u) \leftarrow -f(u, v)$
return f