Maksimalni protok u protocnoj mrezi

Aleksandar Milic

Maj 2013

1 Protocna mreza

Protocna mreza je usmereni graf $G = (V, E)$ sa osobinom da za svaku granu $(u, v) \in E$ imamo unapred odredjeni kapacitet definisan funkcijom $c(u, v) \geq 0$. Ako $(u, v) \in E$ onda $(v, u) \notin E$ i pretpostavljamo da je $c(v, u) = 0$.

Dalje, u mrezi razlikujemo dva posebna cvora: izvor s i slivnik t i pretpostavljamo da $\forall v \in V - \{s, t\}$ postoji put u grafu $(s \ldots v \ldots t)$ sto za posledicu ima to da je graf G povezan i da vazi $|E| \geq |V| - 1$.

Protok u mrezi je funkcija $f : V \times V \to \mathbb{R}$ koja zadovoljava uslove:

Ogranicenje kapaciteta:

$$\Big(\forall u, v \in V\Big)\ 0 \leq f(u, v) \leq c(u, v) \tag{1}$$

Konzervacija protoka:

$$\Big(\forall u, v \in V - \{s, t\}\Big) \sum_{v \in V} f(v, u) = \sum_{v \in V} f(u, v) \tag{2}$$

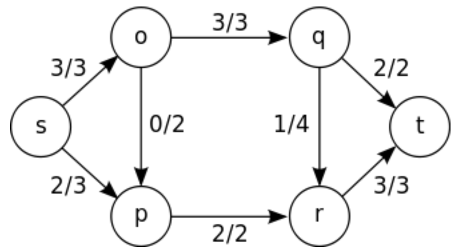
Ako $(u, v) \notin E$, onda u mrezi ne postoji protok iz u u v, pa je $f(u, v) = 0$.

Vrednost protoka u mrezi definisemo kao:

$$|f| = \sum_{v \in V} f(s, v) - \sum_{v \in V} f(v, s) \tag{3}$$

tj. kao razliku ukupnog protoka iz i u izvor s. Uobicajeno je da izvor u protocnoj mrezi nema protok u sebe, pa je u opstem slucaju $\sum_{v \in V} f(v, s) = 0$. Medjutim, uvodjenjem koncepta mreze protocnog reziduuma ova suma ce postati znacajna, jer cemo tada imati i protok ka izvoru.

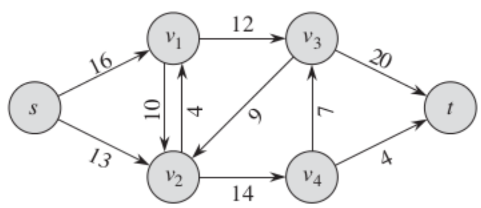
Dakle, u problemu nalazenja maksimalnog protoka u protocnoj mrezi, dat nam je graf $G(V, E)$ sa izvorom s i slivnikom t; gde nam je cilj da pronadjemo maksimalnu vrednost protoka $|f|$.



Slika 1: Protocna mreza

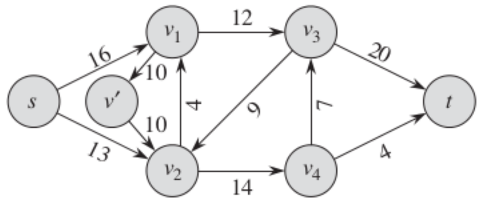
1.1 Graf sa antiparalelnim granama

U slucaju kada u grafu protocne mreze imamo grane $(v_1, v_2) \in E$ i $(v_2, v_1) \in E$, te grane nazivamo antiparalelnim. Posto tada graf ne zadovoljava uslov da ako postoji $(u, v) \in E$ onda $(v, u) \notin E$ cilj nam je da transformisemo graf tako da se zadati uslov ispuni a da transformisani graf bude ekvivalentan originalnom grafu.



Slika 2: Mreza sa antiparalelnim granama

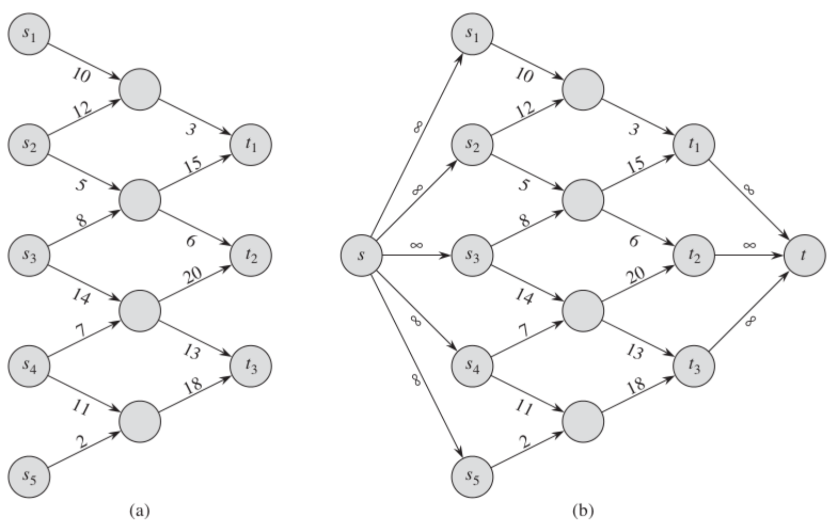
Transformaciju radimo tako sto izaberemo jednu granu iz para antiparalelnih, u ovom slucaju (v_1, v_2) i podelimo je tako sto u graf dodamo cvor v' i zamenimo izabranu granu granama (v_1, v') i (v', v_1) te im dodelimo kapacitet kojeg je imala izabrana grana. Rezultujuci graf je ekvivalentan originalnom, sto se lako pokazuje. !treba dokazati formalno!



Slika 3: Rezultujuci graf nakon transformirasanja

1.2 Mreza sa vise izvora i slivnika

Ako u modelu problema imamo vise izvora i slivnika (slika 4-a), tada nam je zgodno redukovati problem tako da imamo samo jedan izvor s' i jedan slivnik t', koje onda nazivamo super-izvor i super-slivnik. Redukciju radimo tako sto za sve izvore s_i originalnog grafa dodamo grane (s', s_i) sa kapacitetom $c(s', s_i) = \infty$ i za sve slivnike t_i grane (t_i, t') takodje sa neogranicenim kapacitetom. Intuitivno, rezultujuci graf (slika 4-b) je jednak originalnom posto iz super-izvora dolazi neogranicen protok, dok super-slivnik neograniceno trosi protok. !dokazati formalno!



Slika 4: Transformacija grafa sa vise izvora i slivnika

2 Ford-Fulkersonov algoritam za nalazenje maksimalnog protoka

Ford-Fulkersonov algoritam radi tako sto iterativno povecava vrednost protoka. Pocinje sa protokom $\forall u, v \in V f(u, v) = 0$, pa pri svakoj iteraciji dopunjuje protok preko dopunjujuce putanje koju nalazimo u **mrezi reziduuma** G_f sve dok takva putanja u G_f ne postoji. **Max-flow min-cut** teorema nam pokazuje da je dobijeni protok nakon zavrsetka algoritma zaista maksimalni protok u G.

inicijalizujemo protok f na 0;
while *dok postoji putanja dopunjujuceg protoka u* G_f **do**
| dopunjujemo protok f po putanji p
end
return f

2.1 Mreza protocnog reziduuma

Za datu protocnu mrezu $G = (V, E)$ sa izvorom s i slivnikom t definisemo kapacitet reziduuma za $u, v \in V$:

$$c_f(u,v) = \begin{cases} c(u,v) - f(u,v) & (u,v) \in E \\ f(v,u) & (v,u) \in E \\ 0 & inace \end{cases}$$

Mrezu protocnog reziduuma za protocnu mrezu $G = (V, E)$ i funkciju protoka f onda definisemo kao graf $G_f = (V, E_f)$ gde je skup grana E_f dat sa:

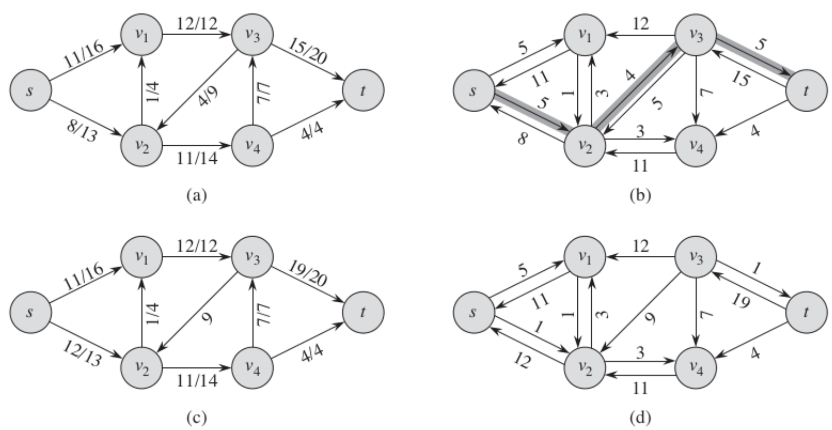
$$E_f = \{(u,v) \in V \times V : c_f(u,v) > 0\}$$

Posto grane u G_f mogu biti ili grane grafa G ili njihove kongruencije suprotnog smera, zakljucujemo da je $|E_f| \leq 2|E|$
Dakle, protocna funkcija u G_f je definisana preko visedelne funkcija c_f.

Dalje, potrebno je definisati funkciju dopunjujuceg protoka koja ce da nam sluzi kao mapa za iterativno povecavanje i smanjivanje protoka u protocnoj mrezi.
Za datu protocnu funkciju f u G i protocnu funkciju mreze reziduuma f' u G_f, definisemo funkciju $f \uparrow f' : V \times V \to \mathbb{R}$ dopunjujuceg protoka f za protok f', kao:

$$(f \uparrow f')(u,v) = \begin{cases} f(u,v) + f'(u,v) - f'(v,u) & (u,v) \in E \\ 0 & inace \end{cases}$$

U prvom slucaju kada $(u, v) \in E$ protok u G mozemo povecati za $f'(u, v)$ po definiciji, ali posto pustanje protoka kroz mrezu reziduuma u protocnoj mrezi predstavlja vracanje protoka, oduzimamo



Slika 5:
(a) Protocna mreza G
(b) Mreza reziduuma sa osencenom putanjom p dopunjujuceg protoka ciji je kapacitet reziduuma $c_f(p) = c_f(v2, v3) = 4$
(c) Protok u G povecan za kapacitet reziduuma putanje p iz G_f
(d) Mreza reziduuma indukovana protokom iz mreze G sa slike (c)

Lema 1
Neka je $G = (V, E)$ protocna mreza sa izvorom s i slivnikom t, i neka je f protok u G, i neka je G_f mreza reziduuma protocne mreze G indukovana protokom f, i neka je f' protok u G_f. Tada je funkcija $f \uparrow f'$ protok u G i vrednost tog protoka definisemo kao $|f \uparrow f'| = |f| + |f'|$.
Dokaz
Prvo treba pokazati da $|f \uparrow f'|$ zadovoljava uslov ogranicenja nad kapacitetom nad svakom granom $(u, v) \in E$ i konzervaciju protoka nad svakim cvorom $v \in V - \{s, t\}$.
Primecujemo da ako $(u, v) \in E$ onda $f(u, v) = c_f(v, u)$. Pa zbog toga, imamo da vazi $f'(v, u) \leq c_f(v, u) = f(u, v)$.

$$\begin{aligned}(f \uparrow f')(u, v) &= f(u, v) + f'(u, v) - f'(v, u) \\ &\geq f(u, v) + f'(u, v) - f(u, v) \\ &= f'(u, v) \\ &\geq 0\end{aligned}$$

Jos nam preostaje da pokazemo da je gornja granica $c(u, v)$

$$\begin{aligned}(f \uparrow f')(u, v) &= f(u, v) + f'(u, v) - f'(v, u) \\ &\leq f(u, v) + f'(u, v) \\ &\leq f(u, v) + c_f(u, v) \\ &= f(u, v) + c(u, v) - f(u, v) \\ &= c(u, v)\end{aligned}$$

Zatim, dokazujemo da $|f \uparrow f'|$ zadovoljava uslov konzervacije protoka. Posto i f i f' zadovoljavaju konzervaciju protoka, imamo da za sve $v \in V - \{s, t\}$

$$\begin{aligned}
&\sum_{v \in V} (f \uparrow f')(u, v) \\
&\qquad = \sum_{v \in V} (f(u, v) + f'(u, v) - f'(v, u)) \\
&\qquad = \sum_{v \in V} f(u, v) + \sum_{v \in V} f'(u, v) - \sum_{v \in V} f'(v, u) \\
&\qquad = \sum_{v \in V} f(v, u) + \sum_{v \in V} f'(v, u) - \sum_{v \in V} f'(u, v) \\
&\qquad = \sum_{v \in V} (f(v, u) + f'(v, u) - f'(u, v)) \\
&\qquad = \sum_{v \in V} (f \uparrow f')(v, u)
\end{aligned}$$

Konacno, preostaje nam da definisemo vrednost protoka $f \uparrow f'$.
Posto u G nemamo antiparalelnih grana (u G_f ih mozemo imati) mozemo da konstruisemo dva zgodna disjunktna skupa cvorova: skup u kome imamo cvorove koji cine grane ka izvoru i skup cvorova koji imaju grane od izvora. $V_1 = \{v : (s, v) \in E\}$ i $V_2 = \{v : (v, s) \in E\}$. Vidimo da vazi $V_1 \cup V_2 \subseteq V$, i zato sto ne dozvoljavamo antiparalelne grane, $V_1 \cap V_2 = \emptyset$

$$\begin{aligned}
&|f \uparrow f'| \\
&\quad = \sum_{v \in V} (f \uparrow f')(s, v) - \sum_{v \in V} (f \uparrow f')(v, s) \\
&\quad = \sum_{v \in V_1} (f \uparrow f')(s, v) - \sum_{v \in V_2} (f \uparrow f')(v, s) \\
&\quad = \sum_{v \in V_1} (f(s, v) + f'(s, v) - f'(v, s)) - \sum_{v \in V_2} (f(v, s) + f'(v, s) - f'(s, v)) \\
&\quad = \sum_{v \in V_1} f(s, v) + \sum_{v \in V_1} f'(s, v) - \sum_{v \in V_1} f'(v, s) \\
&\qquad - \sum_{v \in V_2} f(v, s) + \sum_{v \in V_2} f'(v, s) - \sum_{v \in V_2} f'(s, v) \\
&\quad = \sum_{v \in V_1} f(s, v) - \sum_{v \in V_2} f(v, s) \\
&\qquad + \sum_{v \in V_1} f'(s, v) + \sum_{v \in V_2} f'(s, v) - \sum_{v \in V_1} f'(v, s) - \sum_{v \in V_2} f'(v, s) \\
&\quad = \sum_{v \in V_1} f(s, v) - \sum_{v \in V_2} f(v, s) + \sum_{v \in V_1 \cup V_2} f'(s, v) - \sum_{v \in V_1 \cup V_2} f'(v, s)
\end{aligned} \tag{4}$$

U gornjoj jednacini (1) mozemo da sumiramo nad svim cvorovima u V jer ce za sve ostale cvorove u $V \setminus (V_1 \cup V_2)$ vrednost biti 0. Onda imamo da je vrednost

$$\begin{aligned} &|f \uparrow f'| \\ &\qquad = \sum_{v \in V} f(s,v) - \sum_{v \in V} f(v,s) + \sum_{v \in V} f'(s,v) - \sum_{v \in V} f'(v,s) \\ &\qquad = |f| + |f'| \end{aligned} \tag{5}$$

2.2 Put dopunjujuceg protoka

Za datu protocnu mrezu $G = (V, E)$ i protok nad njom f, put dopunjujuceg protoka p je putanja od s ka t u mrezi reziduuma G_f. Po definiciji mreze reziduuma, protok na grani (u, v) u putanji p mozemo povecati za maksimum $c_f(u, v)$ a da ne prekrsimo uslov ogranicenja nad kapacitetom nad (u, v) ili (v, u) u protocnoj mrezi G.
S toga, maksimalna vrednost s kojom mozemo dopuniti protok na putanji p se onda intuitivno definise kao $c_f(p) = min\{c_f(u, v) : (u, v) \in p\}$

Lema 2
Neka je $G = (V, E)$ protocna mreza, f funkcija protoka nad njom, i p put dopunjujuceg protoka u G_f. Funkciju $f_p : V \times V \to \mathbb{R}$ definisemo

$$f_p(u,v) = \begin{cases} c_f(p) & \text{ako } (u,v) \text{ lezi na putanji } p \\ 0 & \text{inace} \end{cases} \tag{6}$$

Tada je f_p protok u G_f sa vrednoscu $|f_p| = c_f(p) > 0$

Lema **(2)** za posledicu ima to da ako dopunimo protok f protokom f_p, dobicemo novi protok u G cija je vrednost veca od f tj. bliza je maksimalnom protoku u G od f.

Posledica 1
Neka je $G = (V, E)$ protocna mreza, f funkcija protoka nad njom, p put u G_f dopunjujuceg protoka, i f_p protok definisan u **(6)**
Tada je $f \uparrow f_p$ protok u G sa vrednoscu $|f \uparrow f_p| = |f| + |f_p| > |f|$
Dokaz se lako izvodi preko lema **(1)** i **(2)**.

2.3 Rezovi protocne mreze

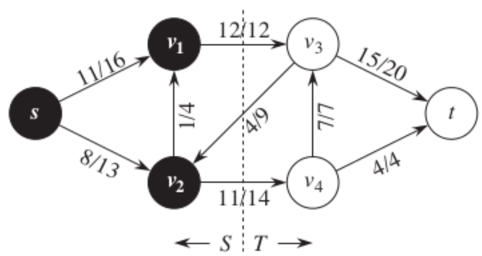
Da bismo dokazali **Max-flow min-cut** teoremu potrebno je da definisemo pojam reza nad protocnom mrezom. Rez (S, T) protocne mreze $G = (V, E)$ je particija skupa V u skupove S i $T = V \setminus S$ tako da $s \in S$ i $t \in T$. Ako je f protok u G onda je ukupan protok $f(S, T)$ preko reza (S, T)

$$f(S,T) = \sum_{u \in S} \sum_{v \in T} f(u,v) - \sum_{u \in S} \sum_{v \in T} f(v,u) \tag{7}$$

Kapacitet reza (S, T) definisemo kao

$$c(S,T) = \sum_{u \in S} \sum_{v \in T} c(u,v) \tag{8}$$

Minimalni rez nad protocnom mrezom G je rez ciji je kapacitet minimalan nad svim rezovima u G



Slika 6: Rez (S, T) protocne mreze sa slike **(3)**, gde je $S = \{s, v_1, v_2\}$ i $T = \{v_3, v_4, t\}$. Cvorovi u S su prikazani u crnoj, a cvorovi u T u beloj boji. Ukupan protok preko (S, T) je $f(S, T) = 19$ a kapacitet reza je $c(S, T) = 26$

Lema 3
Neka je f protok u G sa izvorom s i slivnikom t, i neka je (S, T) neki rez nad G. Onda je ukupan protok nad rezom (S, T) dat sa $f(S, T) = |f|$

Dokaz
Jednacinu konzervacije protoka **(2)** za $\forall u \in V - \{s, t\}$ mozemo zapisati kao

$$\sum_{v \in V} f(u,v) - \sum_{v \in V} f(v,u) = 0$$

Ako uzmemo definiciju $|f|$ iz jednacine **(3)** i saberemo je sa levom stranom gornje jednacine koja ima vrednost 0 za svaki cvor iz $S - \{s\}$, dobijamo

$$\begin{aligned}
&|f| \\
&\quad = \sum_{v \in V} f(s,v) - \sum_{v \in V} f(v,s) + \sum_{u \in S-\{s\}} \left(\sum_{v \in V} f(u,v) - \sum_{v \in V} f(v,u) \right) \\
&\quad = \sum_{v \in V} f(s,v) - \sum_{v \in V} f(v,s) + \sum_{u \in S-\{s\}} \sum_{v \in V} f(u,v) - \sum_{u \in S-\{s\}} \sum_{v \in V} f(v,u) \\
&\quad = \sum_{v \in V} \left(f(s,v) + \sum_{u \in S-\{s\}} f(u,v) \right) - \sum_{v \in V} \left(f(v,s) + \sum_{u \in S-\{s\}} f(v,u) \right) \\
&\quad = \sum_{v \in V} \sum_{u \in S} f(u,v) - \sum_{v \in V} \sum_{u \in S} f(v,u)
\end{aligned}$$

Posto $V = S \cup T$ i $S \cap T = \emptyset$, mozemo da sume nad V predstavimo kao sume nad S i T, pa imamo

$$\begin{aligned}
|f| &= \sum_{v \in S} \sum_{u \in S} f(u,v) + \sum_{v \in T} \sum_{u \in S} f(u,v) - \sum_{v \in S} \sum_{u \in S} f(v,u) - \sum_{v \in T} \sum_{u \in S} f(v,u) \\
&= \sum_{v \in T} \sum_{u \in S} f(u,v) - \sum_{v \in T} \sum_{u \in S} f(v,u) + \left(\sum_{v \in S} \sum_{u \in S} f(u,v) - \sum_{v \in S} \sum_{u \in S} \right)
\end{aligned}$$

Sume u zagradi se ponistavaju jer se svaki izraz $f(u,v)$ za svako $u, v \in S$ pojavljuje jednom i u jednoj i u drugoj sumi, pa onda imamo

$$\begin{aligned}
&|f| \\
&\quad = \sum_{u \in S} \sum_{v \in T} f(u,v) - \sum_{u \in S} \sum_{v \in T} f(v,u) \\
&\quad = f(S,T);
\end{aligned}$$

Posledica leme **(3)** nam omogucuje da postavimo gornju granicu nad vrednoscu protoka preko **kapaciteta reza**

Posledica 2
Vrednost bilo kog protoka f u protocnoj mrezi G je ogranicena odozgo kapacitetom bilo kog reza nad G

Dokaz
Neka je (S, T) neki rez nad G i neka je f neki protok nad G, po lemi **(3)** i po nejednacini ogranicenja kapaciteta vidimo da vazi

$$\begin{aligned}
|f| &= f(S,T) \\
&= \sum_{u \in S} \sum_{v \in T} f(u,v) - \sum_{u \in S} \sum_{v \in T} f(v,u) \\
&\leq \sum_{u \in S} \sum_{v \in T} f(u,v) \\
&\leq \sum_{u \in S} \sum_{v \in T} c(u,v) \\
&= c(S,T)
\end{aligned}$$

Teorema 1 (Max-flow min-cut)
Ako je f protok u protocnoj mrezi $G = (V, E)$ sa izvorom s i slivnikom t, onda su sledeci uslovi ekvivalentni:

1. f je maksimalan protok u G
2. Mreza reziduuma G_f ne sadrzi putanju dopunjujuceg protoka
3. $|f| = c(S,T)$ za neki rez (S,T) nad G

Dokaz
(1) $\Rightarrow$ (2):
Pretpostavimo suprotno, tj. da je f maksimalni protok u G ali da G_f sadrzi putanju dopunjujuceg protoka p. Tada, zbog posledice **(1)**, vidimo da postoji protok f dopunjen protokom f_p cija je vrednost striktno veca od $|f|$. Sto je kontradikcija sa uzetom pretpostavkom.
(2) $\Rightarrow$ (3):
Pretpostavimo da G_f ne sadrzi putanju dopunjujuceg protoka od s do t. Definisimo skupove $S = \{v \in V$: postoji put od s do v u $G_f\}$ i $T = V - S$. Tada je particija (S, T) rez nad G jer vazi $s \in S$ i $t \notin T$ jer po pretpostavci ne postaji putanja od s do t u G_f. Posmatrajmo par cvorova $u \in S$ i $v \in T$. Ako $(u, v) \in E$, onda $f(u, v) = c(u, v)$ jer bi inace $(u, v) \in E_f$ pa bi v bio u skupu S. Ako $(v, u) \in E$ onda $f(v, u) = 0$ jer bi inace imali $c_f(u, v) = f(v, u) > 0$ i $(u, v) \in E_f$ pa bi v bio u skupu S. Naravno, da (u, v) i (v, u) nisu u E, vazilo bi $f(u, v) = f(v, u) = 0$, pa zato:

$$\begin{aligned}
&f(S,T) \\
&\quad= \sum_{u \in S} \sum_{v \in T} f(u,v) - \sum_{v \in T} \sum_{u \in S} f(v,u) \\
&\quad= \sum_{u \in S} \sum_{v \in T} c(u,v) - \sum_{v \in T} \sum_{u \in S} 0 \\
&\quad= c(S,T)
\end{aligned}$$

(3) $\Rightarrow$ (1):
Sledi iz posledice **(2)**, jer $|f| \leq c(S,T)$ za sve rezove (S,T) i uslova da je $|f| = c(S,T)$ sto implicira da je f maksimalni protok u G

3 Implementacija Ford-Fulkerson algoritma

```python
class Edge(object):
    def __init__(self, u, v, w):
        self.source = u
        self.sink = v
        self.capacity = w
    def __repr__(self):
        return "%s->%s:%s" % (self.source, self.sink, self.capacity)

class FlowNetwork(object):
    def __init__(self):
        self.adj = {}
        self.flow = {}

    def add_vertex(self, vertex):
        self.adj[vertex] = []

    def get_edges(self, v):
        return self.adj[v]

    def add_edge(self, u, v, w=0):
        if u == v:
            raise ValueError("u == v")
        edge = Edge(u,v,w)
        redge = Edge(v,u,0)
        edge.redge = redge
        redge.redge = edge
        self.adj[u].append(edge)
        self.adj[v].append(redge)
        self.flow[edge] = 0
        self.flow[redge] = 0

    def find_path(self, source, sink, path):
        if source == sink:
            return path
        for edge in self.get_edges(source):
            residual = edge.capacity - self.flow[edge]
            if residual > 0 and not (edge,residual) in path:
                result = self.find_path( edge.sink, sink, path +
                    [(edge,residual)] )
                if result != None:
                    return result
```

```
    def max_flow(self, source, sink):
        path = self.find_path(source, sink, [])
        while path != None:
            flow = min(res for edge,res in path)
            for edge,res in path:
                self.flow[edge] += flow
                self.flow[edge.redge] -= flow

            path = self.find_path(source, sink, [])

        return sum(self.flow[edge] for edge in self.get_edges(source))


g = FlowNetwork();
map(g.add_vertex, ['s','o','p','q','r','t']);
g.add_edge('s','o',3)
g.add_edge('s','p',3)
g.add_edge('o','p',2)
g.add_edge('o','q',3)
g.add_edge('p','r',2)
g.add_edge('r','t',3)
g.add_edge('q','r',4)
g.add_edge('q','t',2)
print g.max_flow('s','t')
```

References

[1] Thomas H. Cormen. *Introduction to Algorithms.* The MIT Press, 2009.