Gráfalgoritmusok - Folyamfeladatok

Gaskó Noémi

2023. április 16.

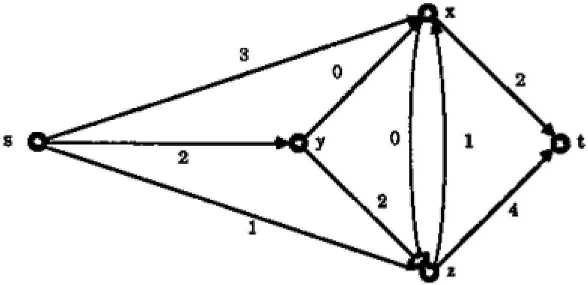
Tartalomjegyzék

Egy feladat

Sherlock Holmes ördöglakatokat rendel Kolozsvárról Londonba. Megkéri néhány ismerõsét, hogy vigyék el neki, de mindenki csak néhány darabot tud elszállítani, adott városokig, amit a következő ábra szemléltet:

Feladat (folyt.)

Kérdés: hány ördöglakatot kaphat meg Sherlock Londonban?



Egyéb feladatok

Tengeri szállítás

Adott m kikötõ, x_1,x_2,...,x_m, amelyekben bizonyos áru rendre s_1, s_2,...,s_m mennyiségben fordul elõ; n kikötõ, y_1, y_2, ...,y_n, ahol az illetõ áruból rendre d_1,d_2,...,d_n mennyiséget igényelnek.

Egyéb feladatok

Tengeri szállítás

Adott m kikötõ, x_1,x_2,...,x_m, amelyekben bizonyos áru rendre s_1, s_2,...,s_m mennyiségben fordul elõ; n kikötõ, y_1, y_2, ...,y_n, ahol az illetõ áruból rendre d_1,d_2,...,d_n mennyiséget igényelnek.

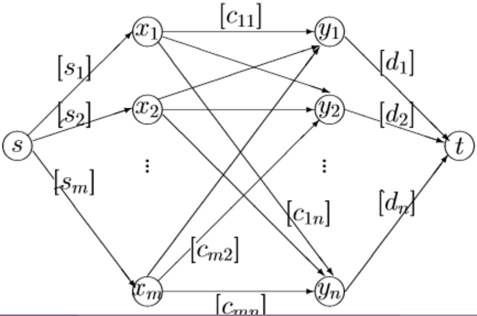
Hogyan lehet megoldani minél több áru elszállítását, ha tudjuk, hogy egy adott x_i kikötõbõl egy y_j kikötõtõbe adott intervallumban c_{ij} mennyiség szállítható (a hajó kapacitása).

Egyéb feladatok

Tengeri szállítás

Adott m kikötõ, x_1,x_2,...,x_m, amelyekben bizonyos áru rendre s_1, s_2,...,s_m mennyiségben fordul elõ; n kikötõ, y_1, y_2, ...,y_n, ahol az illetõ áruból rendre d_1,d_2,...,d_n mennyiséget igényelnek.

Hogyan lehet megoldani minél több áru elszállítását, ha tudjuk, hogy egy adott x_i kikötõbõl egy y_j kikötõtõbe adott intervallumban c_{ij} mennyiség szállítható (a hajó kapacitása).



Egyéb feladatok (2)

Családi kirándulás

a_1, a_2,...,a_m családok rendre s_1,s_2,...,s_m tagúak
n busz áll a rendelkezésünkre: b_1, b_2,...,b_n, amelyek rendre d_1, d_2, ..., d_n személyt szállítanak

Egyéb feladatok (2)

Családi kirándulás

a_1, a_2,...,a_m családok rendre s_1,s_2,...,s_m tagúak

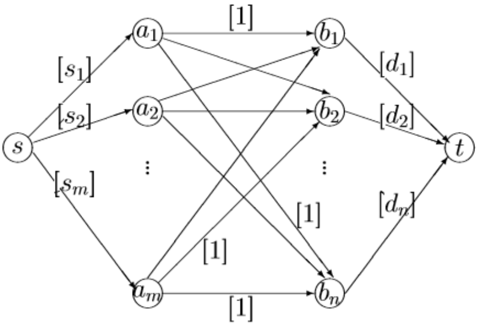
n busz áll a rendelkezésünkre: b_1, b_2,...,b_n, amelyek rendre d_1, d_2, ..., d_n személyt szállítanak

Lehetséges-es úgy megszervezni a kirándulást, hogy a családtagok mind különbözõ buszokba kerüljenek?

Egyéb feladatok (2)

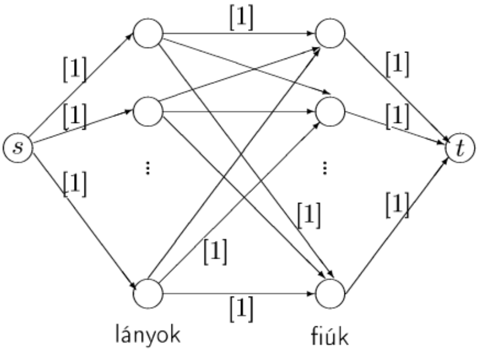
Családi kirándulás

a_1, a_2,...,a_m családok rendre s_1,s_2,...,s_m tagúak
n busz áll a rendelkezésünkre: b_1, b_2,...,b_n, amelyek rendre d_1, d_2, ..., d_n személyt szállítanak
Lehetséges-es úgy megszervezni a kirándulást, hogy a családtagok mind különbözõ buszokba kerüljenek?



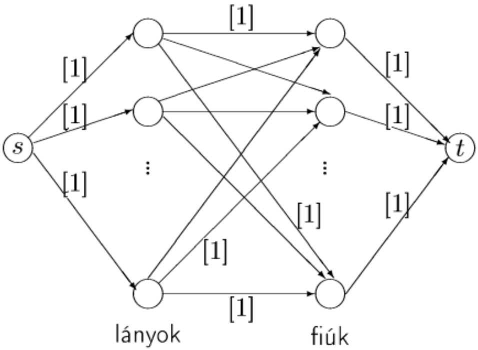
Egyéb feladatok (3)

Buli Meg lehet-e szervezni egy táncot úgy, hogy minden lány olyan fiúval táncoljon, akit ismer?



Egyéb feladatok (3)

Buli Meg lehet-e szervezni egy táncot úgy, hogy minden lány olyan fiúval táncoljon, akit ismer?



Akkor van megoldás, ha létezik olyan maximális folyam, ami telíti az s-bõl kifutó éleket.

Egyéb feladatok (4)

- repülõgépek útvonalainak és repülésének ütemezése
- baseball meccseken a kiesések ütemezése
- ....

Folyamok - szállítási hálózatok

(Közlekedési) hálózat

Egy közlekedési hálózat egy (V, E) irányított gráf a következõ tulajdonságokkal:

- $\exists s \in V : N^{be}(s) = \emptyset$, s forrás,
- $\exists t \in V : N^{ki}(s) = \emptyset$, t nyelő,
- a kapacitásfüggvényt a következőképpen: $\alpha : V \times V \to \mathbb{Z}_+$ úgy hogy $\alpha(x, y) > 0$, ha $(x, y) \in E$ $\alpha(x, y) = 0$, ha $(x, y) \notin E$

A hálózatat a következőképpen jelöljük: $H = (V, E, \alpha, s, t)$.

Értelmezések

Folyam

Folyamnak nevezzük az $f : V \times V \to \mathbb{Z}_+$ függvényt, amely a következő tulajdonságokkal rendelkezik:

- $f(x, y) \leq \alpha(x, y)$ (kapacitás-megszorítás)
- $f(x, V) = f(V, x) \forall x \in V - \{s, t\}$ (egyensúly feltétel)

Értelmezések

Folyam

Folyamnak nevezzük az $f : V \times V \to \mathbb{Z}_+$ függvényt, amely a következő tulajdonságokkal rendelkezik:

- $f(x, y) \leq \alpha(x, y)$ (kapacitás-megszorítás)
- $f(x, V) = f(V, x) \forall x \in V - \{s, t\}$ (egyensúly feltétel)

Folyam értéke

A $v(f) = f(s, V)$ értéket az f folyam értékének nevezzük. Egy folyam telít egy élt, ha azon az élen a folyam értéke egyező a kapacitással.

Értelmezések

Folyam

Folyamnak nevezzük az $f : V \times V \to \mathbb{Z}_+$ függvényt, amely a következő tulajdonságokkal rendelkezik:

- $f(x, y) \leq \alpha(x, y)$ (kapacitás-megszorítás)
- $f(x, V) = f(V, x) \forall x \in V - \{s, t\}$ (egyensúly feltétel)

Folyam értéke

A $v(f) = f(s, V)$ értéket az f folyam értékének nevezzük. Egy folyam telít egy élt, ha azon az élen a folyam értéke egyező a kapacitással.

Tétel

$v(f) = f(s, V) = f(V, t)$ bármely f folyamra.

A vágat

Vágat értelmezése

Ha egy hálózatban van a csúcsoknak egy olyan $A \subseteq V, \overline{A} = V - A$ partíciója, hogy $s \in A$ és $t \in \overline{A}$, akkor ezt $(A, \overline{A})$ vágatnak nevezzük.

A vágat

Vágat értelmezése

Ha egy hálózatban van a csúcsoknak egy olyan $A \subseteq V, \overline{A} = V - A$ particiója, hogy $s \in A$ és $t \in \overline{A}$, akkor ezt $(A, \overline{A})$ vágatnak nevezzük.

Kapacitás

Az $(A, \overline{A})$ vágat kapacitása egyenlõ az éleihez rendelt kapacitások összegével:

$$\alpha(A, \overline{A}) = \sum_{x \in A, y \in \overline{A}} \alpha(x, y).$$

Minimális vágat

A legkisebb kapacitású vágatot egy adott hálózatban minimális vágatnak nevezzük.

Minimális vágat

A legkisebb kapacitású vágatot egy adott hálózatban minimális vágatnak nevezzük.

Tétel

Ha $(A, \overline{A})$ vágat egy hálózatban, akkor tetszõleges f folyamra:

$$v(f) = f(A, \overline{A}) - f(\overline{A}, A) \leq \alpha(A, \overline{A}).$$

Minimális vágat

A legkisebb kapacitású vágatot egy adott hálózatban minimális vágatnak nevezzük.

Tétel

Ha $(A, \overline{A})$ vágat egy hálózatban, akkor tetszõleges f folyamra:

$$v(f) = f(A, \overline{A}) - f(\overline{A}, A) \leq \alpha(A, \overline{A}).$$

Ford-Fulkerson tétele

Egy hálózatban egy maximális folyam értéke egyenlő a minimális vágatkapacitással.

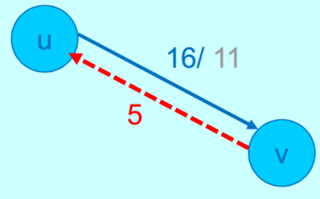
Reziduális gráf

Reziduális kapacitás:

$$c_f(u,v) = \alpha(u,v) - f(u,v)$$

Példa:

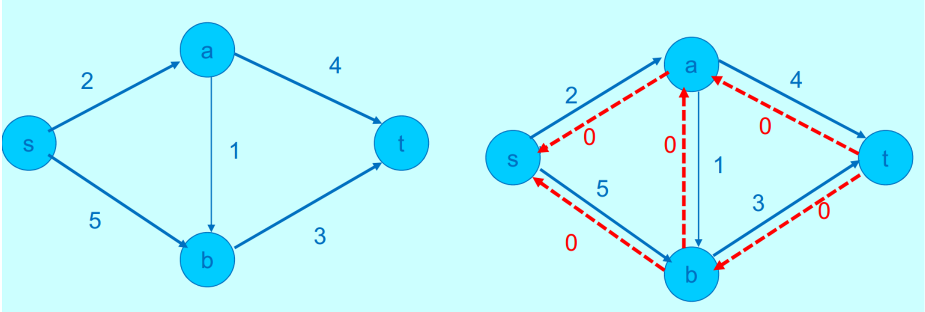
$$\alpha(u,v) = 16, f(u,v) = 11 \rightarrow c_f(u,v) = 5$$



Reziduális gráf

Reziduális gráf

$G_f = (V, E_f)$ reziduális hálózat, ahol
$E_f = \{(u, v) \in V \times V : C_f(u, v) > 0\}$



Ford-Fulkerson algoritmus

Ford-Fulkerson(G, s, t)

1. **for** minden $(u, v) \in E$ él esetén
2. $\quad f(u, v) = f(v, u) = 0$
3. **while** létezik p út s-bol t-be a G_f reziduális gráfban
4. $\quad c_f(p) = min\{c_f(u, v) : (u, v) \in p\}$
5. $\quad$ **for** minden $(u, v) \in p$ esetén
6. $\quad\quad f(u, v) = f(u, v) + c_f(p)$
7. $\quad\quad f(v, u) = -f(u, v)$

Egy példa

Lásd 7_jegyzet.pdf

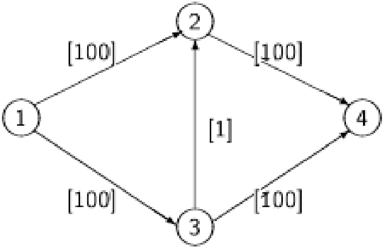
Ford-Fulkerson algoritmus elemzése

Mi történik valós kapacitás esetén?

lásd 7_jegyzet.pdf

Ford-Fulkerson algoritmus elemzése

Hány lépés alatt találjuk meg a maximális folyamot?



Ford-Fulkerson algoritmus elemzése (2)

- két lépésben is megkapható: 1-2-4 útvonalon növeljük 100-al, majd az 1-3-4-en is növelhetjük 100-al.

Ford-Fulkerson algoritmus elemzése (2)

- két lépésben is megkapható: 1-2-4 útvonalon növeljük 100-al, majd az 1-3-4-en is növelhetjük 100-al.
- de ha elõször az 1-3-2-4 utat választjuk 1-el növelhetjük, ezek után az 1-2-3-4 láncot használva a folyamot 1-el lehet növelni, így 200 lépésben kapjuk meg a folyamot.

Edmonds-Karp algoritmus

- a Ford-Fulkerson algoritmuson alapul
- alapötlet: a lehetsége láncok közül a legrövidebbet (a legkevesebb élből állót) választjuk - ha ezt szélességi bejárással határozzuk meg akkor a legrövidebbet kapjuk

Boykov-Kolmogorov algoritmus

- a Ford-Fulkerson algoritmuson alapul, képfeldolgozás esetén szokták használni
- két keresõ fa: egyik a forrástól, másik a nyelõtõl, ezek változnak az algoritmus futása során, és ezek tartják nyilván a javító utakat

Pumpáló algoritmusok

- az előző algoritmusok esetén javító utak, azaz érvényes folyamok
- a pumpáló algoritmusban nincs érvényes folyam, addig módosul az előfolyam ameddig érvényessé nem válik
- előfolyam: nemnegaitivitás, kapacitáskorlát, a folyammegmaradás kritériumát nem kell betartsa

Pumpáló algoritmus

ELOFOLYAM_INIT(G,s,t)

```
\\ inicializálási lépés f(u,v) és h(u,v), ∀u,v ∈ V
for minden v ∈ V do
    v.h = 0
    v.e = 0
for minden (u,v) ∈ E do
    (u,v).f=0
s.h = |V|
for minden v ∈ s.Adj do
    (s,v).f = c(s,v)
    v.e = c(s,v)
    s.e = s.e − c(s,v)
```

Pumpáló algoritmus (folyt.)

POMPALAS(u, v)

\\\\ alkalmazzuk ha: $u \notin \{s, t\} \wedge u.e > 0 \wedge c_f(u, v) > 0 \wedge u.h = v.h + 1$
\\\\ pompálja a folyam mennyiséget $\Delta_f(u, v) = min(u.e, c_f(u, v))$
$\Delta_f(u, v) = min(u.e, c_f(u, v))$
if $(u, v) \in E$ **then**
    $(u, v).f = (u, v).f + \Delta_f(u, v)$
else
    $(v, u).f = (v, u).f - \Delta_f(u, v)$
$u.e = u.e - \Delta_f(u, v)$
$v.e = v.e + \Delta_f(u, v)$

Pumpáló algoritmus (folyt.)

EMELES(u)

```
\\ alkalmazzuk, ha:
```
$u \notin \{s,t\} \wedge u.e > 0 \wedge [u.h \leq v.h | \forall v \in V, (u,v) \in E_f]$

```
\\ megnöveli a magasságot
```
$u.h$

$u.h = 1 + min\{v.h | (u,v) \in E_f\}$

POMPALAS_ELOFOLYAM(G,s,t)

ELOFOLYAM_INIT(G,s,t)

while TRUE **do**

    if $\exists u \notin \{s,t\} \wedge u.e > 0 \wedge c_f(u,v) > 0 \wedge u.h = v.h + 1$ **then**

        POMPALAS(u,v)

        continue

    if $\exists u \notin \{s,t\} \wedge u.e > 0 \wedge [u.h \leq v.h | \forall v \in V, (u,v) \in E_f]$ **then**

        EMELES(u)

        continue

    break

Topológiai pompálás

TOPOLOGIAI_POMPALAS(G,s,t)

ELOFLYAM_INICIALIZALAS(G,s,t)
$L = V \setminus \{s, t\}$
for minden $u \in V \setminus \{s, t\}$ **do**
    $u.curent = u.N.head$
$u = L.head$
while $u \neq NIL$ **do**
    regi_magassag $= u.h$
    LETOLTES(u)
    if $u.h >$ regi_magassag **then**
        u-t tegyük át az L lista elejére
    $u.next$

Topológiai pompálás (folyt.)

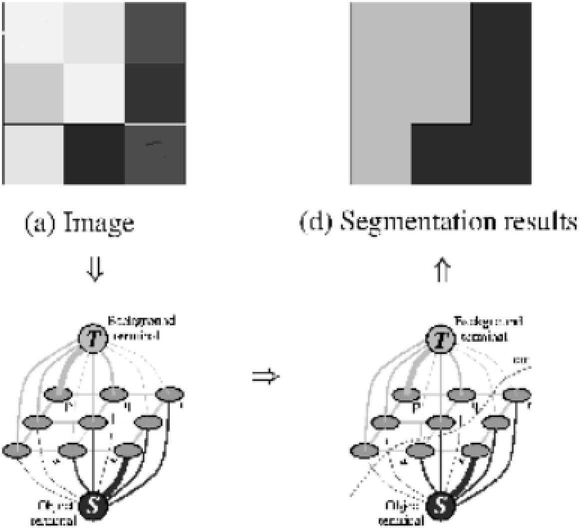
LETOLTES(u)

while $u.e > 0$ **do**
    $v = u.curent$
    if $v == NIL$ **then**
        EMELES(u)
        $u.curent = u.N.head$
    else if $c_f(u, v) > 0 \wedge u.h == v.h + 1$ **then**
        POMPALAS(u,v)
    else
        $u.curent = u.kovetkezo\_szomszed$

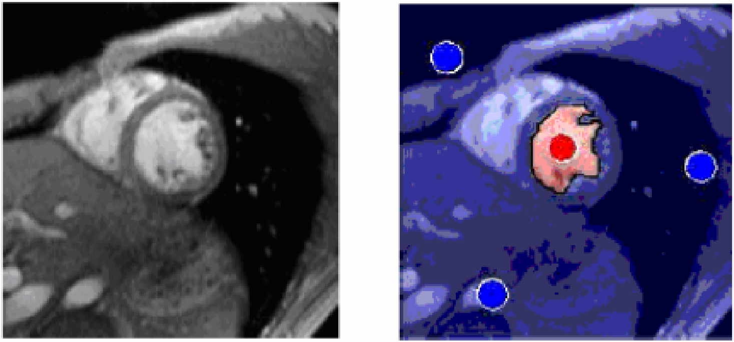
Algoritmusok bonyolultsága

- Ford-Fulkerson: $O(|E||f_{max}|)$
- Edmonds-Karp: $O(|E|^2|V|)$
- Pumpáló algoritmus: $O(|V|^2E)$
- Topológia pumpáló algoritmus: $O(|V|^3)$

Minimális vágások alkalmazásai



Minimális vágások alkalmazásai (2)



Original image A minimum cut

Általánosított folyam

Általánosítás

Hálózat általánosítása: a kapacitás mellett megadunk egy alsó határt is minden élen, a folyam értéke a két érték közé kell essen.

Általánosított folyam

Általánosítás

Hálózat általánosítása: a kapacitás mellett megadunk egy alsó határt is minden élen, a folyam értéke a két érték közé kell essen.

Értelemzés

Legyen $H^* = (V, E, \beta, \alpha, s, t)$ általánosított hálózat, ahol V,E,s,t jelentése ugyanaz. A β és α függvények $V \times V$-n értelmezettek és nem negatív egész értékûek. Ezek alsó és felsõ kapacitásfüggvények, $\beta \leq \alpha$ és

1. $\alpha(x, y) > 0$ ha $(x, y) \in E$
2. $\alpha(x, y) = 0$ ha $(x, y) \notin E$.

Általánosított folyam

Az általánosított folyam egy általánosított hálózatban az $f : V \times V \to \mathbb{Z}_+$ függvény, amely a következő tulajdonságokkal rendelkezik:

- $\beta(x, y) \leq f(x, y) \leq \alpha(x, y)$, ha $x, y \in V$
- $f(V, x) = f(x, V)$, ha $x \in V - \{s, t\}$.

Általánosított folyam

Az általánosított folyam egy általánosított hálózatban az $f : V \times V \to \mathbb{Z}_+$ függvény, amely a következő tulajdonságokkal rendelkezik:

- $\beta(x, y) \leq f(x, y) \leq \alpha(x, y)$, ha $x, y \in V$
- $f(V, x) = f(x, V)$, ha $x \in V - \{s, t\}$.

Általánosított folyam értéke

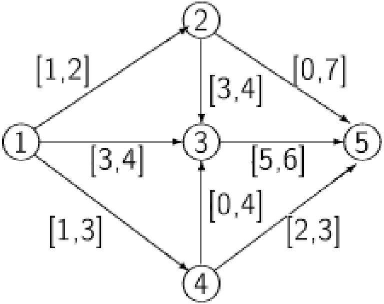
Az általánosított folyam értéke $v(f) = f(s, V) = f(V, t)$

Általánosított folyam értéke

Létezik-e mindig maximális folyam?

Általánosított folyam értéke

Létezik-e mindig maximális folyam? Milyen feltétel mellett létezik maximális folyam?



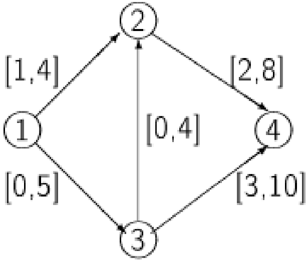
Ebben a hálózatban létezhet-e maximális folyam?

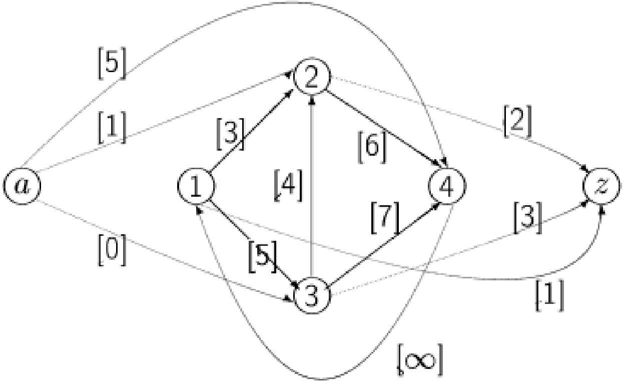
Általánosított folyam átalakítása hagyományos folyamfeladattá

A H általánosított hálózathoz hozzárendelünk egy hagyományos $H^* = (V^*, E^*, \alpha^*, a, z)$ a következőképpen:

- $V^* = V \cup \{(a, z)\}$
- $E^* = E \cup \{(a, x) | N_G^{be}(x) \neq \emptyset\} \cup \{(x, z) | N_G^{ki}(x) \neq \emptyset\} \cup \{(t, s)\}$
- $\alpha^*(x, y) = \alpha(x, y) - \beta(x, y), \forall x, y \in V.$
- $\alpha^*(a, x) = \beta(V, x)$
- $\alpha^*(x, z) = \beta(x, V)$
- $\alpha^*(t, s) = \infty$

Általánosított folyam átalakítása hagyományos folyamfeladattá

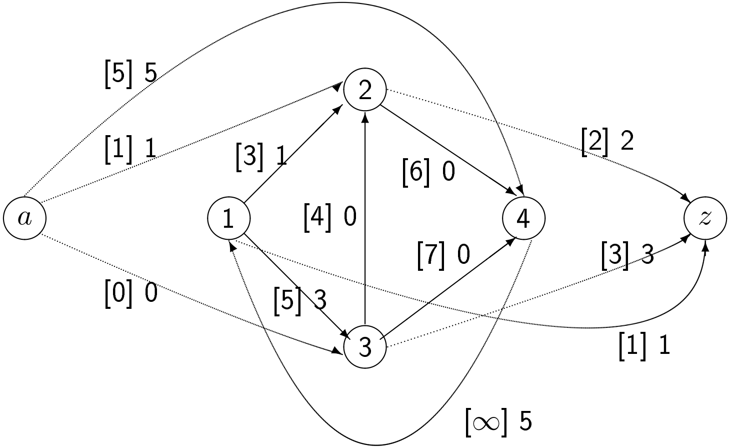



[5]
2
[1]
[3]
[2]
[6]
a
1
[4]
4
z
[7]
[3]
[0]
[5]
3
[1]
[∞]

Mikor létezik általánosított folyam?

A

H általánosított hálózatban akkor és csakis akkor létezik általánosított folyam, ha a H^* hozzárendelt hálózatban létezik olyan maximális folyam, amely telíti az a csúcsból kiinduló éleket.

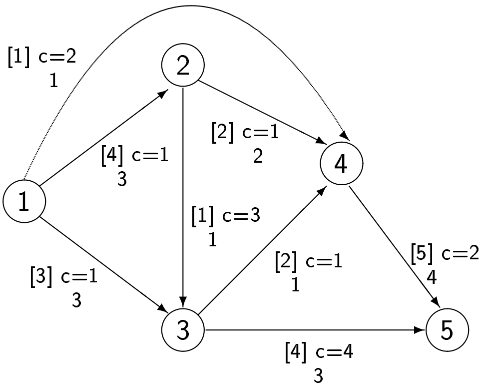

[5] 5
[1] 1
[3] 1
[6] 0
[2] 2
[4] 0
[7] 0
[3] 3
[0] 0
[5] 3
[1] 1
[∞] 5
a
1
2
3
4
z

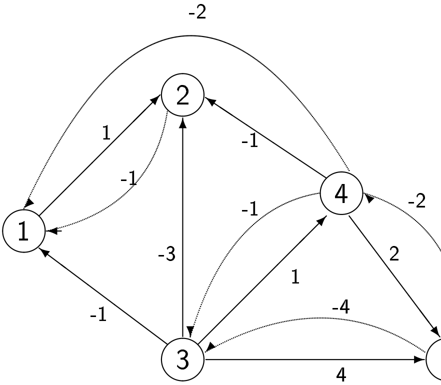
Minimális költségű maximális folyam

Legyen adott egy $G = (V, E, \alpha, s, t)$ hálózat és egy $c : V \times V \to R_+$ költségfüggvény (ahol R_+ a nem negatív valós számok halmaza). Ha f folyam G-ben, akkor az f folyam *költsége*:

$$\mathcal{C}(f) := \sum_{x,y \in V} c(x, y) f(x, y)$$

Ha $(x, y) \notin E$, akkor vehetjük, hogy $c(x, y) = 0$.
Természetesen, a legkisebb költségű maximális folyam érdekel bennünket.
Hogyan lehet eldönteni egy maximális folyamról, hogy minimális költségű?
Megoldás: egy adott hálózathos és maximális folyamhoz hozzárendelünk egy súlyozott irányított gráfot, amelyben a negatív hosszúságú kör hiánya mutatja meg, hogy a maximális folyam minimális költségű.





Ennek a folyamnak a költsége
$\mathcal{C}(f) = 3 \cdot 1 + 3 \cdot 1 + 2 \cdot 1 + 1 \cdot 3 + 1 \cdot 2 + 1 \cdot 1 + 3 \cdot 4 + 4 \cdot 2 = 34.$

Több forrás, több nyelő

- visszavezethető egy klasszikus folyamfeladatra
- hozzáadunk egy szuperforrást és egy szupernyelőt, és a szuperforrást összekötjük az összes forrással ∞ kapacitású éllel, illetve az eredeti nyelőket a szupernyelővel ∞ kapacitású éllel

Kapacitás a csúcsokon

- a csúcsoknak is van egy értéke, ami azt jelzi, hogy egy korlátozás van
- ebben az esetben megduplázzuk a csúcsokat, és közéjük egy élt teszünk, amely kapacitása egyenlő lesz a csúcs kapacitásával
- azok az élek, amelyek eddig a csomópontra mutattak, a "bemenő" csomópontba fognak mutatni, míg amik kifelé mutattak a másik csomópontból fognak kifelé mutatni

Forrásanyag

- Kása jegyzet
- Jean Claude Fournier, Graph Theory and Applications, 2009
- Santana Sahu Ray, Graph Theory with Algorithms and its Applications, Springer, 2013