Kombinatorikus optimalizálás

Hálózati folyamok, páros gráfok párosításai és élszínezései

2023. március 7.

ZH időpontok

- 1. ZH: április 19 (szerda) 18-20
- 1. pZH: május 3 (szerda) 18-20
- 2. ZH: május 18 (csütörtök) 18-20
- 2. pZH: június 5 (pótlási hét hétfő) 18-20

ZH időpontok

- 1. ZH: április 19 (szerda) 18-20
- 1. pZH: május 3 (szerda) 18-20
- 2. ZH: május 18 (csütörtök) 18-20
- 2. pZH: június 5 (pótlási hét hétfő) 18-20

Változás!

A március 21-diki előadás az STFKis teremben lesz.

A hidegháború hatása a kombinatorikus optimalizálásra

- Az 1950-es években járunk. Valódi veszélynek tűnik, hogy a szovjet hadsereg tankjai megindulnak nyugat-Európa ellen.

A hidegháború hatása a kombinatorikus optimalizálásra

- Az 1950-es években járunk. Valódi veszélynek tűnik, hogy a szovjet hadsereg tankjai megindulnak nyugat-Európa ellen.
- A haditechnikát vasúton szállítják, ezért a vasút elleni légi támadás látszik az legalkalmasabb védekezésnek.

A hidegháború hatása a kombinatorikus optimalizálásra

- Az 1950-es években járunk. Valódi veszélynek tűnik, hogy a szovjet hadsereg tankjai megindulnak nyugat-Európa ellen.
- A haditechnikát vasúton szállítják, ezért a vasút elleni légi támadás látszik az legalkalmasabb védekezésnek.
- A csapásmérés optimalizálásához használt gráf csúcsai a vasúti igazgatóságok, élei pedig a vasúti kapcsolatok voltak. Ismert volt az egyes vasútvonalak ezer tonnában mért kapacitása is. Ezt a gráfot kellett úgy kettévágni, hogy ne maradjon kapcsolat a szovjet támaszpontok és nyugat-Európa között.

A hidegháború hatása a kombinatorikus optimalizálásra

- Az 1950-es években járunk. Valódi veszélynek tűnik, hogy a szovjet hadsereg tankjai megindulnak nyugat-Európa ellen.
- A haditechnikát vasúton szállítják, ezért a vasút elleni légi támadás látszik az legalkalmasabb védekezésnek.
- A csapásmérés optimalizálásához használt gráf csúcsai a vasúti igazgatóságok, élei pedig a vasúti kapcsolatok voltak. Ismert volt az egyes vasútvonalak ezer tonnában mért kapacitása is. Ezt a gráfot kellett úgy kettévágni, hogy ne maradjon kapcsolat a szovjet támaszpontok és nyugat-Európa között.
- Egy titkos jelentésben Ford és Fulkerson olyan algoritmust írnak le, ami tetszőleges gráfon megoldja ezt a problémát: megtalál néhány vasútvonalat, melyeket elvágva a feltétel teljesül és ezen belül az összkapacitásuk a lehető legkisebb.

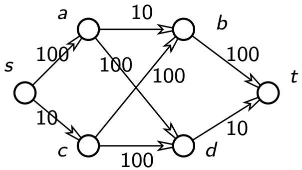
A hidegháború hatása a kombinatorikus optimalizálásra

- Az 1950-es években járunk. Valódi veszélynek tűnik, hogy a szovjet hadsereg tankjai megindulnak nyugat-Európa ellen.
- A haditechnikát vasúton szállítják, ezért a vasút elleni légi támadás látszik az legalkalmasabb védekezésnek.
- A csapásmérés optimalizálásához használt gráf csúcsai a vasúti igazgatóságok, élei pedig a vasúti kapcsolatok voltak. Ismert volt az egyes vasútvonalak ezer tonnában mért kapacitása is. Ezt a gráfot kellett úgy kettévágni, hogy ne maradjon kapcsolat a szovjet támaszpontok és nyugat-Európa között.
- Egy titkos jelentésben Ford és Fulkerson olyan algoritmust írnak le, ami tetszőleges gráfon megoldja ezt a problémát: megtalál néhány vasútvonalat, melyeket elvágva a feltétel teljesül és ezen belül az összkapacitásuk a lehető legkisebb.
- Az jelentésből az is kiderül, hogy a sértetlen hálózat ugyanilyen összkapacitással képes nyugatra szállítani a hadianyagot.

A hidegháború hatása a kombinatorikus optimalizálásra

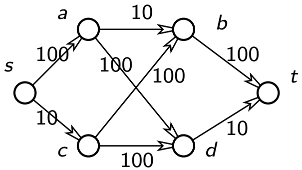
- Az 1950-es években járunk. Valódi veszélynek tűnik, hogy a szovjet hadsereg tankjai megindulnak nyugat-Európa ellen.
- A haditechnikát vasúton szállítják, ezért a vasút elleni légi támadás látszik az legalkalmasabb védekezésnek.
- A csapásmérés optimalizálásához használt gráf csúcsai a vasúti igazgatóságok, élei pedig a vasúti kapcsolatok voltak. Ismert volt az egyes vasútvonalak ezer tonnában mért kapacitása is. Ezt a gráfot kellett úgy kettévágni, hogy ne maradjon kapcsolat a szovjet támaszpontok és nyugat-Európa között.
- Egy titkos jelentésben Ford és Fulkerson olyan algoritmust írnak le, ami tetszőleges gráfon megoldja ezt a problémát: megtalál néhány vasútvonalat, melyeket elvágva a feltétel teljesül és ezen belül az összkapacitásuk a lehető legkisebb.
- Az jelentésből az is kiderül, hogy a sértetlen hálózat ugyanilyen összkapacitással képes nyugatra szállítani a hadianyagot.
- A jelentés titkosítását 1999-ben oldották fel.

Maximális nagyságú folyamok (ismétlés)



Def: **Hálózat**: (G, s, t, c) négyes, ahol $G = (V, E)$ irányított gráf, $s, t \in V$ **terminálok** (termelő, fogyasztó), $c : E \to \mathbb{R}_+$ pedig a **kapacitásfüggvény**.

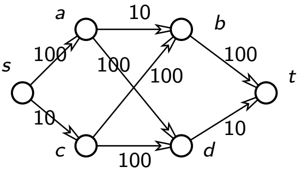
Maximális nagyságú folyamok (ismétlés)



Def: **Hálózat**: (G, s, t, c) négyes, ahol $G = (V, E)$ irányított gráf, $s, t \in V$ **terminálok** (termelő, fogyasztó), $c : E \to \mathbb{R}_+$ pedig a **kapacitásfüggvény**.

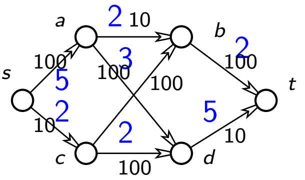
Megj: (1) Az ,,eredeti" problémában s a szovjet tankok forrása (keleten), t pedig a megvédeni kívánt terület (nyugaton).
(2) Az ,,eredeti" problémában a G gráf irányítatlan, itt irányított. Később látni fogjuk, hogy az irányítatlan gráfokhoz tartozó problémák irányított gráfon is megfogalmazhatók, ezért a fenti modell általánosabb a feladatot motiválónál.
(3) Rendszerint s forrás, t pedig nyelő a folyamproblémát leíró gráfban, de ennek nem szükséges feltétlenül így lennie.

Maximális nagyságú folyamok (ismétlés)



Def: **Hálózat**: (G, s, t, c) négyes, ahol $G = (V, E)$ irányított gráf, $s, t \in V$ **terminálok** (termelő, fogyasztó), $c : E \to \mathbb{R}_+$ pedig a **kapacitásfüggvény**.

Maximális nagyságú folyamok (ismétlés)

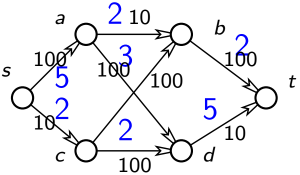


Def: **Hálózat**: (G, s, t, c) négyes, ahol $G = (V, E)$ irányított gráf, $s, t \in V$ **terminálok** (termelő, fogyasztó), $c : E \to \mathbb{R}_+$ pedig a **kapacitásfüggvény**.

st-**folyam**: olyan $x : E \to \mathbb{R}_+$ függvény, amire teljesül a **kapacitasfeltétel**, azaz $0 \leq x(e) \leq c(e) \forall e \in E$, és a **Kirchhoff-szabály**, miszerint tetsz. nemterminális $v \neq s, t$ csúcsra $\sum\{x(uv) : uv \in E\} = \sum\{x(vz) : vz \in E\}$.

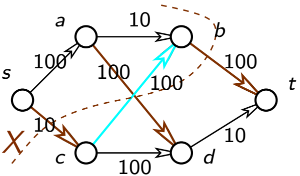
Az x *st*-**folyam nagysága** az s-ből kilépő *nettó* folyammennyiség: $m_x = \sum\{x(sv) : sv \in E\} - \sum\{x(zs) : zs \in E\}$.

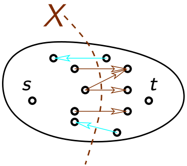
Maximális nagyságú folyamok (ismétlés)



Def: **Hálózat**: (G, s, t, c) négyes, ahol $G = (V, E)$ irányított gráf, $s, t \in V$ **terminálok** (termelő, fogyasztó), $c : E \to \mathbb{R}_+$ pedig a **kapacitásfüggvény**.

Maximális nagyságú folyamok (ismétlés)





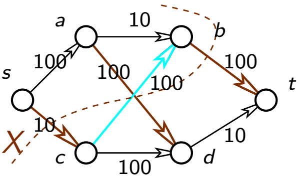
Def: **Hálózat**: (G, s, t, c) négyes, ahol $G = (V, E)$ irányított gráf, $s, t \in V$ **terminálok** (termelő, fogyasztó), $c : E \to \mathbb{R}_+$ pedig a **kapacitásfüggvény**.

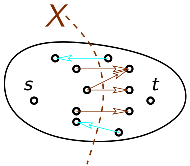
indukált *st*-vágás: $s \in X \subseteq V - t$ esetén az X és $V - X$ között futó élek halmaza.

A fenti ***st*-vágás kapacitása**: $c(X) = \sum\{c(uv) : u \in X, v \notin X\}$.

Megj: Egy *st*-vágás kapacitásába tehát csak az X-et elhagyó élek számítanak, az X-be belépők nem.

Maximális nagyságú folyamok (ismétlés)



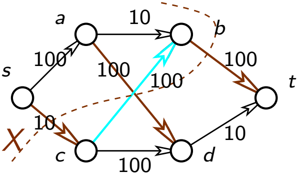


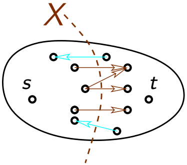
Def: **Hálózat**: (G, s, t, c) négyes, ahol $G = (V, E)$ irányított gráf, $s, t \in V$ **terminálok** (termelő, fogyasztó), $c : E \to \mathbb{R}_+$ pedig a **kapacitásfüggvény**.

indukált *st*-vágás: $s \in X \subseteq V - t$ esetén az X és $V - X$ között futó élek halmaza.

A fenti ***st*-vágás kapacitása**: $c(X) = \sum\{c(uv) : u \in X, v \notin X\}$.

Maximális nagyságú folyamok (ismétlés)





Def: **Hálózat**: (G, s, t, c) négyes, ahol $G = (V, E)$ irányított gráf, $s, t \in V$ **terminálok** (termelő, fogyasztó), $c : E \to \mathbb{R}_+$ pedig a **kapacitásfüggvény**.

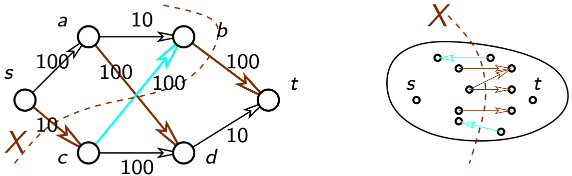
indukált *st*-vágás: $s \in X \subseteq V - t$ esetén az X és $V - X$ között futó élek halmaza.

A fenti ***st*-vágás kapacitása**: $c(X) = \sum\{c(uv) : u \in X, v \notin X\}$.

Lemma: Ha x *st*-folyam és $s \in X \not\ni t$, akkor $m_x \leq c(X)$. $\square$

Megj: A Lemma azt mondja ki, hogy egyetlen *st*-folyam nagysága sem haladhatja meg egyetlen *st*-vágás kapacitását sem.

Maximális nagyságú folyamok (ismétlés)



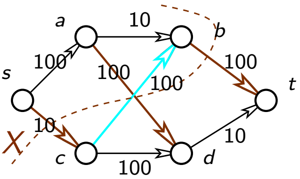
Def: **Hálózat**: (G, s, t, c) négyes, ahol $G = (V, E)$ irányított gráf, $s, t \in V$ **terminálok** (termelő, fogyasztó), $c : E \to \mathbb{R}_+$ pedig a **kapacitásfüggvény**.

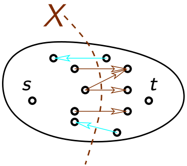
indukált ***st*-vágás**: $s \in X \subseteq V - t$ esetén az X és $V - X$ között futó élek halmaza.

A fenti ***st*-vágás kapacitása**: $c(X) = \sum\{c(uv) : u \in X, v \notin X\}$.

Lemma: Ha x st-folyam és $s \in X \not\ni t$, akkor $m_x \leq c(X)$. $\square$

Maximális nagyságú folyamok (ismétlés)





Def: **Hálózat**: (G, s, t, c) négyes, ahol $G = (V, E)$ irányított gráf, $s, t \in V$ **terminálok** (termelő, fogyasztó), $c : E \to \mathbb{R}_+$ pedig a **kapacitásfüggvény**.

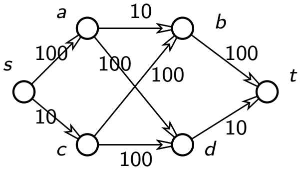
indukált *st*-vágás: $s \in X \subseteq V - t$ esetén az X és $V - X$ között futó élek halmaza.

A fenti ***st*-vágás kapacitása**: $c(X) = \sum\{c(uv) : u \in X, v \notin X\}$.

Lemma: Ha x st-folyam és $s \in X \not\ni t$, akkor $m_x \leq c(X)$. $\square$

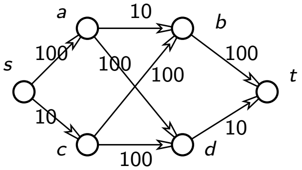
Köv: Ha $m_x = c(X)$ teljesül egy x st-folyam és egy $s \in X \subseteq V - X$ esetén, akkor x egy maximális nagyságú st-folyam és X minimális kapacitású st-vágást indukál. $\square$

A Ford-Fulkerson-algoritmus (ismétlés)



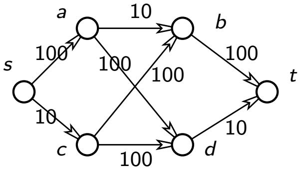
Ford-Fulkerson (MFMC) tétel: Tetsz. (G, s, t, c) hálózatban található olyan x st-folyam és $s \in X \subseteq V - X$, amire $m_x = c(X)$.
Megj: Azaz az st-folyamok maximális nagysága megegyezik az st-vágások minimális kapacitásával.

A Ford-Fulkerson-algoritmus (ismétlés)



Ford-Fulkerson (MFMC) tétel: Tetsz. (G, s, t, c) hálózatban található olyan x st-folyam és $s \in X \subseteq V - X$, amire $m_x = c(X)$.

A Ford-Fulkerson-algoritmus (ismétlés)



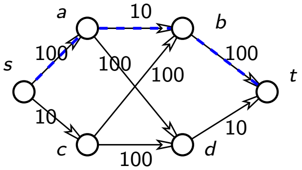
Ford-Fulkerson (MFMC) tétel: Tetsz. (G, s, t, c) hálózatban található olyan x st-folyam és $s \in X \subseteq V - X$, amire $m_x = c(X)$.

Biz: Javító utas algoritmus

Az $x \equiv 0$ folyamból kiindulva mindig javító utak mentén küldünk folyamot s-ből t-be, amíg csak tudunk.

Telítetlen élekből álló st-úton mindig tudunk folyamot növelni.

A Ford-Fulkerson-algoritmus (ismétlés)



Ford-Fulkerson (MFMC) tétel: Tetsz. (G, s, t, c) hálózatban található olyan x st-folyam és $s \in X \subseteq V - X$, amire $m_x = c(X)$.

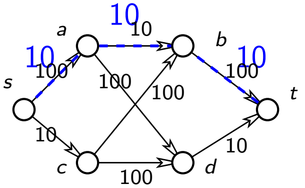
Biz: Javító utas algoritmus

Az $x \equiv 0$ folyamból kiindulva mindig javító utak mentén küldünk folyamot s-ből t-be, amíg csak tudunk.

Telítetlen élekből álló st-úton mindig tudunk folyamot növelni.

A Ford-Fulkerson-algoritmus (ismétlés)

$sabt$: 10



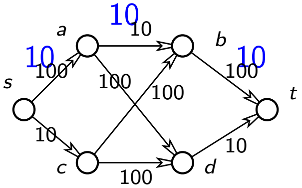
Ford-Fulkerson (MFMC) tétel: Tetsz. (G, s, t, c) hálózatban található olyan x st-folyam és $s \in X \subseteq V - X$, amire $m_x = c(X)$.

Biz: Javító utas algoritmus

Az $x \equiv 0$ folyamból kiindulva mindig javító utak mentén küldünk folyamot s-ből t-be, amíg csak tudunk.

Telítetlen élekből álló st-úton mindig tudunk folyamot növelni.

A Ford-Fulkerson-algoritmus (ismétlés)



sabt : 10

Ford-Fulkerson (MFMC) tétel: Tetsz. (G, s, t, c) hálózatban található olyan x st-folyam és $s \in X \subseteq V - X$, amire $m_x = c(X)$.

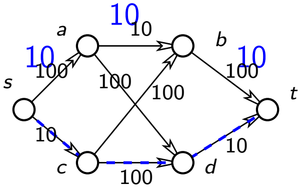
Biz: Javító utas algoritmus

Az $x \equiv 0$ folyamból kiindulva mindig javító utak mentén küldünk folyamot s-ből t-be, amíg csak tudunk.

Telítetlen élekből álló st-úton mindig tudunk folyamot növelni.

A Ford-Fulkerson-algoritmus (ismétlés)

$sabt$: 10



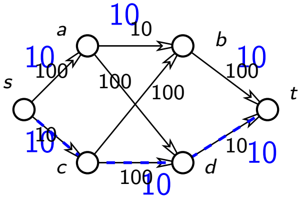
Ford-Fulkerson (MFMC) tétel: Tetsz. (G, s, t, c) hálózatban található olyan x st-folyam és $s \in X \subseteq V - X$, amire $m_x = c(X)$.

Biz: Javító utas algoritmus

Az $x \equiv 0$ folyamból kiindulva mindig javító utak mentén küldünk folyamot s-ből t-be, amíg csak tudunk.

Telítetlen élekből álló st-úton mindig tudunk folyamot növelni.

A Ford-Fulkerson-algoritmus (ismétlés)



sabt : 10

scdt : 10

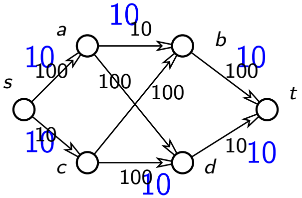
Ford-Fulkerson (MFMC) tétel: Tetsz. (G, s, t, c) hálózatban található olyan x st-folyam és $s \in X \subseteq V - X$, amire $m_x = c(X)$.

Biz: Javító utas algoritmus

Az $x \equiv 0$ folyamból kiindulva mindig javító utak mentén küldünk folyamot s-ből t-be, amíg csak tudunk.

Telítetlen élekből álló st-úton mindig tudunk folyamot növelni.

A Ford-Fulkerson-algoritmus (ismétlés)



$sabt$: 10

$scdt$: 10

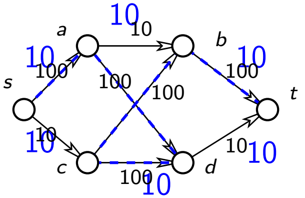
Ford-Fulkerson (MFMC) tétel: Tetsz. (G, s, t, c) hálózatban található olyan x st-folyam és $s \in X \subseteq V - X$, amire $m_x = c(X)$.

Biz: Javító utas algoritmus

Az $x \equiv 0$ folyamból kiindulva mindig javító utak mentén küldünk folyamot s-ből t-be, amíg csak tudunk.

Telítetlen élekből álló st-úton mindig tudunk folyamot növelni.

A Ford-Fulkerson-algoritmus (ismétlés)



sabt : 10

scdt : 10

Ford-Fulkerson (MFMC) tétel: Tetsz. (G, s, t, c) hálózatban található olyan x st-folyam és $s \in X \subseteq V - X$, amire $m_x = c(X)$.

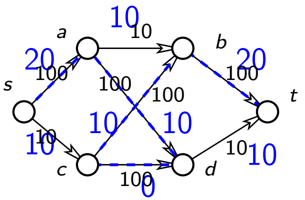
Biz: Javító utas algoritmus

Az $x \equiv 0$ folyamból kiindulva mindig javító utak mentén küldünk folyamot s-ből t-be, amíg csak tudunk.

Telítetlen élekből álló st-úton mindig tudunk folyamot növelni.

Titkos fegyver: ha nincs ilyen út, akkor (a telítetlen éleken történő folyamnövelés mellett) a pozitív élen csökkenthetjük a folyamot. Ezáltal visszafelé küldünk virtuális folyamot egy él mentén. Ha így találunk st-utat, akkor annak a mentén is tudunk folyamot növelni.

A Ford-Fulkerson-algoritmus (ismétlés)



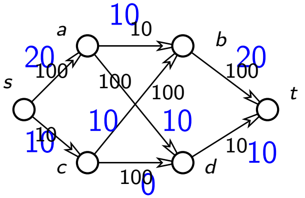
sabt : 10
scdt : 10
sadcbt : 10

Ford-Fulkerson (MFMC) tétel: Tetsz. (G, s, t, c) hálózatban található olyan x st-folyam és $s \in X \subseteq V - X$, amire $m_x = c(X)$.

Biz: Javító utas algoritmus

Az $x \equiv 0$ folyamból kiindulva mindig javító utak mentén küldünk folyamot s-ből t-be, amíg csak tudunk.
Telítetlen élekből álló st-úton mindig tudunk folyamot növelni.
Titkos fegyver: ha nincs ilyen út, akkor (a telítetlen éleken történő folyamnövelés mellett) a pozitív élen csökkenthetjük a folyamot. Ezáltal visszafelé küldünk virtuális folyamot egy él mentén. Ha így találunk st-utat, akkor annak a mentén is tudunk folyamot növelni.

A Ford-Fulkerson-algoritmus (ismétlés)



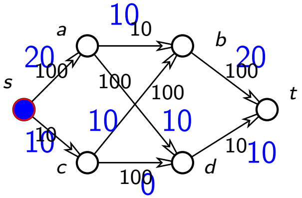
$sabt$: 10
$scdt$: 10
$sadcbt$: 10
$m_x = 30$

Ford-Fulkerson (MFMC) tétel: Tetsz. (G, s, t, c) hálózatban található olyan x st-folyam és $s \in X \subseteq V - X$, amire $m_x = c(X)$.

Biz: Javító utas algoritmus

Az $x \equiv 0$ folyamból kiindulva mindig javító utak mentén küldünk folyamot s-ből t-be, amíg csak tudunk.
Telítetlen élekből álló st-úton mindig tudunk folyamot növelni.
Titkos fegyver: ha nincs ilyen út, akkor (a telítetlen éleken történő folyamnövelés mellett) a pozitív élen csökkenthetjük a folyamot. Ezáltal visszafelé küldünk virtuális folyamot egy él mentén. Ha így találunk st-utat, akkor annak a mentén is tudunk folyamot növelni.

A Ford-Fulkerson-algoritmus (ismétlés)



$sabt$: 10
$scdt$: 10
$sadcbt$: 10
$m_x = 30$

Ford-Fulkerson (MFMC) tétel: Tetsz. (G, s, t, c) hálózatban található olyan x st-folyam és $s \in X \subseteq V - X$, amire $m_x = c(X)$.

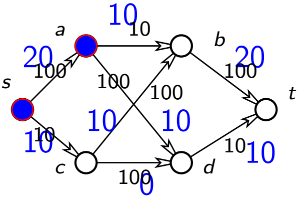
Biz: Javító utas algoritmus

Az $x \equiv 0$ folyamból kiindulva mindig javító utak mentén küldünk folyamot s-ből t-be, amíg csak tudunk.

Telítetlen élekből álló st-úton mindig tudunk folyamot növelni.

Titkos fegyver: ha nincs ilyen út, akkor (a telítetlen éleken történő folyamnövelés mellett) a pozitív élen csökkenthetjük a folyamot. Ezáltal visszafelé küldünk virtuális folyamot egy él mentén. Ha így találunk st-utat, akkor annak a mentén is tudunk folyamot növelni.

A Ford-Fulkerson-algoritmus (ismétlés)



sabt : 10

scdt : 10

sadcbt : 10

$m_x = 30$

Ford-Fulkerson (MFMC) tétel: Tetsz. (G, s, t, c) hálózatban található olyan x st-folyam és $s \in X \subseteq V - X$, amire $m_x = c(X)$.

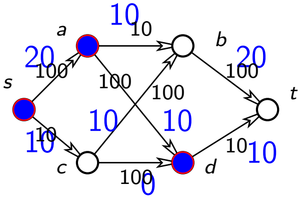
Biz: Javító utas algoritmus

Az $x \equiv 0$ folyamból kiindulva mindig javító utak mentén küldünk folyamot s-ből t-be, amíg csak tudunk.

Telítetlen élekből álló st-úton mindig tudunk folyamot növelni.

Titkos fegyver: ha nincs ilyen út, akkor (a telítetlen éleken történő folyamnövelés mellett) a pozitív élen csökkenthetjük a folyamot. Ezáltal visszafelé küldünk virtuális folyamot egy él mentén. Ha így találunk st-utat, akkor annak a mentén is tudunk folyamot növelni.

A Ford-Fulkerson-algoritmus (ismétlés)



$sabt : 10$
$scdt : 10$
$sadcbt : 10$
$m_x = 30$
$X = \{s, a, d\}$

Ford-Fulkerson (MFMC) tétel: Tetsz. (G, s, t, c) hálózatban található olyan x st-folyam és $s \in X \subseteq V - X$, amire $m_x = c(X)$.

Biz: Javító utas algoritmus

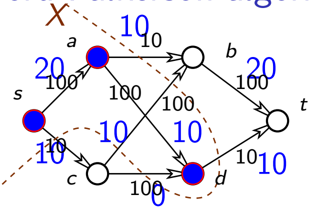
Az $x \equiv 0$ folyamból kiindulva mindig javító utak mentén küldünk folyamot s-ből t-be, amíg csak tudunk.

Telítetlen élekből álló st-úton mindig tudunk folyamot növelni.

Titkos fegyver: ha nincs ilyen út, akkor (a telítetlen éleken történő folyamnövelés mellett) a pozitív élen csökkenthetjük a folyamot.

Ezáltal visszafelé küldünk virtuális folyamot egy él mentén. Ha így találunk st-utat, akkor annak a mentén is tudunk folyamot növelni.

A Ford-Fulkerson-algoritmus (ismétlés)



$sabt$: 10
$scdt$: 10
$sadcbt$: 10
$m_x = 30$
$X = \{s, a, d\}$
$c(X) = 30$

Ford-Fulkerson (MFMC) tétel: Tetsz. (G, s, t, c) hálózatban található olyan x st-folyam és $s \in X \subseteq V - X$, amire $m_x = c(X)$.

Biz: Javító utas algoritmus

Az $x \equiv 0$ folyamból kiindulva mindig javító utak mentén küldünk folyamot s-ből t-be, amíg csak tudunk.
Telítetlen élekből álló st-úton mindig tudunk folyamot növelni.
Titkos fegyver: ha nincs ilyen út, akkor (a telítetlen éleken történő folyamnövelés mellett) a pozitív élen csökkenthetjük a folyamot.
Ezáltal visszafelé küldünk virtuális folyamot egy él mentén. Ha így találunk st-utat, akkor annak a mentén is tudunk folyamot növelni.
Ha nincs javító út, akkor a telítetlen ill. megfordított pozitív éleken s-ből elérhető csúcsok X halmazára $m_x = c(X)$ teljesül. $\square$

Egész kapacitások, általánosított hálózatok

Egészértékűségi (EgÉr) lemma: Ha a (G, s, t, c) hálózatban minden $c(e)$ élkapacitás egész szám, akkor van olyan maximális nagyságú x folyam, amire $x(e)$ is egész szám minden e élre. $\square$

Egész kapacitások, általánosított hálózatok

Egészértékűségi (EgÉr) lemma: Ha a (G, s, t, c) hálózatban minden $c(e)$ élkapacitás egész szám, akkor van olyan maximális nagyságú x folyam, amire $x(e)$ is egész szám minden e élre. $\square$

Biz: A javító utas algoritmus során mindvégig igaz marad, hogy $x(e)$ egész szám a hálózat minden e élére. Ezért ez az algoritmus végén kapott maximális nagyságú folyamra is teljesül. $\square$

Egész kapacitások, általánosított hálózatok

Egészértékűségi (EgÉr) lemma: Ha a (G, s, t, c) hálózatban minden $c(e)$ élkapacitás egész szám, akkor van olyan maximális nagyságú x folyam, amire $x(e)$ is egész szám minden e élre. $\square$

Egész kapacitások, általánosított hálózatok

Egészértékűségi (EgÉr) lemma: Ha a (G, s, t, c) hálózatban minden $c(e)$ élkapacitás egész szám, akkor van olyan maximális nagyságú x folyam, amire $x(e)$ is egész szám minden e élre. $\square$

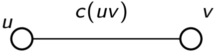
A folyamprobléma általánosításai

Egész kapacitások, általánosított hálózatok

Egészértékűségi (EgÉr) lemma: Ha a (G, s, t, c) hálózatban minden $c(e)$ élkapacitás egész szám, akkor van olyan maximális nagyságú x folyam, amire $x(e)$ is egész szám minden e élre. $\square$

A folyamprobléma általánosításai

Irányítatlan élek kezelése:

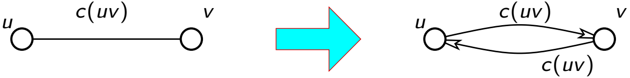


Egész kapacitások, általánosított hálózatok

Egészértékűségi (EgÉr) lemma: Ha a (G, s, t, c) hálózatban minden $c(e)$ élkapacitás egész szám, akkor van olyan maximális nagyságú x folyam, amire $x(e)$ is egész szám minden e élre. $\square$

A folyamprobléma általánosításai

Irányítatlan élek kezelése:

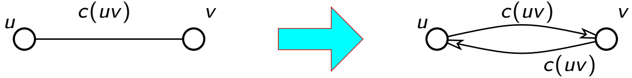


Egész kapacitások, általánosított hálózatok

Egészértékűségi (EgÉr) lemma: Ha a (G, s, t, c) hálózatban minden $c(e)$ élkapacitás egész szám, akkor van olyan maximális nagyságú x folyam, amire $x(e)$ is egész szám minden e élre. □

A folyamprobléma általánosításai

Irányítatlan élek kezelése:



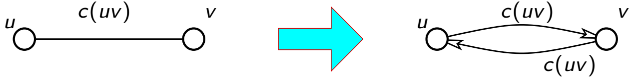
Több termelő, több fogyasztó, egyféle termék:

Egész kapacitások, általánosított hálózatok

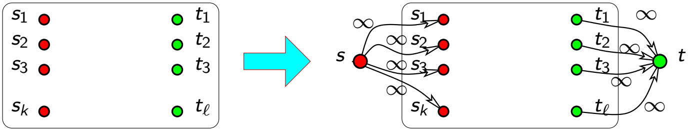
Egészértékűségi (EgÉr) lemma: Ha a (G, s, t, c) hálózatban minden $c(e)$ élkapacitás egész szám, akkor van olyan maximális nagyságú x folyam, amire $x(e)$ is egész szám minden e élre. $\square$

A folyamprobléma általánosításai

Irányítatlan élek kezelése:



Több termelő, több fogyasztó, egyféle termék:

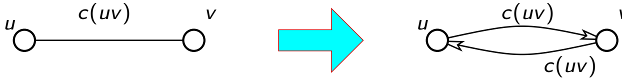


Egész kapacitások, általánosított hálózatok

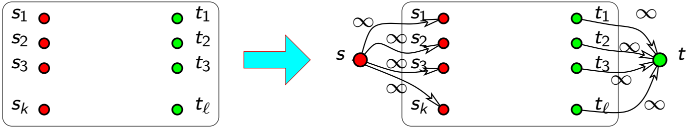
Egészértékűségi (EgÉr) lemma: Ha a (G, s, t, c) hálózatban minden $c(e)$ élkapacitás egész szám, akkor van olyan maximális nagyságú x folyam, amire $x(e)$ is egész szám minden e élre. $\square$

A folyamprobléma általánosításai

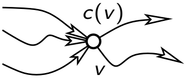
Irányítatlan élek kezelése:



Több termelő, több fogyasztó, egyféle termék:



Csúcskapacitások bevezetése akkor indokolt, ha a hálózat egyes csúcsainak korlátozott az átbocsátóképessége.

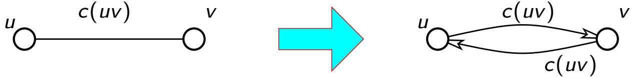


Egész kapacitások, általánosított hálózatok

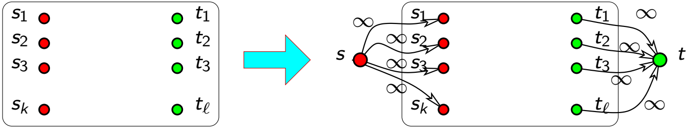
Egészértékűségi (EgÉr) lemma: Ha a (G, s, t, c) hálózatban minden $c(e)$ élkapacitás egész szám, akkor van olyan maximális nagyságú x folyam, amire $x(e)$ is egész szám minden e élre. $\square$

A folyamprobléma általánosításai

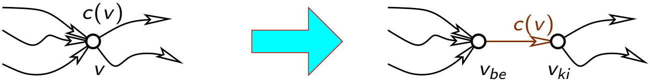
Irányítatlan élek kezelése:



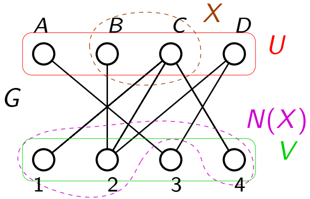
Több termelő, több fogyasztó, egyféle termék:



Csúcskapacitások bevezetése akkor indokolt, ha a hálózat egyes csúcsainak korlátozott az átbocsátóképessége.



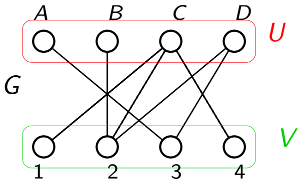
Páros gráfok (ismétlés)



Ez itt egy **páros gráf**, **színosztály**ai U és V. **Párosítás** alatt független éleket értünk, amelyeknek nincs közös végpontja. A **teljes párosítás** olyan párosítás, ami G minden csúcsát lefedi. Az $X \subseteq U$ csúcshalmaz **szomszédság**a az $N(X) \subseteq V$ csúcshalmaz.

Frobenius tétele: Az U és V színosztályokból álló páros gráfnak pontosan akkor van teljes párosítása, ha $|U| = |V|$ és teljesül a Hall-feltétel, azaz $|X| \leq |N(X)|$ minden $X \subseteq U$ részhalmazra. $\square$

Páros gráfok (ismétlés)

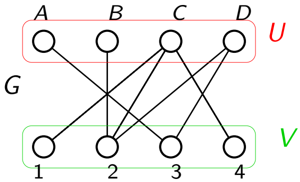


Ez itt egy **páros gráf**, **színosztály**ai U és V. **Párosítás** alatt független éleket értünk, amelyeknek nincs közös végpontja. A **teljes párosítás** olyan párosítás, ami G minden csúcsát lefedi. Az $X \subseteq U$ csúcshalmaz **szomszédság**a az $N(X) \subseteq V$ csúcshalmaz.

Frobenius tétele: Az U és V színosztályokból álló páros gráfnak pontosan akkor van teljes párosítása, ha $|U| = |V|$ és teljesül a Hall-feltétel, azaz $|X| \leq |N(X)|$ minden $X \subseteq U$ részhalmazra. □

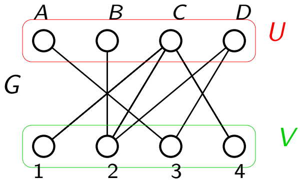
Kőnig-tétel: Ha G páros gráf, akkor $\nu(G) = \tau(G)$, azaz a független élek maximális száma megegyezik a lefogó ponthalmaz minimális méretével. □

Páros gráfok (ismétlés)



Ez itt egy **páros gráf**, **színosztály**ai U és V. **Párosítás** alatt független éleket értünk, amelyeknek nincs közös végpontja. A **teljes párosítás** olyan párosítás, ami G minden csúcsát lefedi. Az $X \subseteq U$ csúcshalmaz **szomszédság**a az $N(X) \subseteq V$ csúcshalmaz.

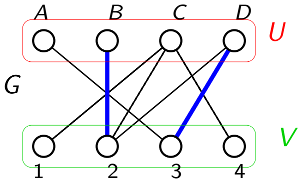
Páros gráfok (ismétlés)



	A	B	C	D
1	0	0	1	0
2	0	1	1	1
3	1	0	0	1
4	0	0	1	0

Ez itt egy **páros gráf**, **színosztály**ai U és V. **Párosítás** alatt független éleket értünk, amelyeknek nincs közös végpontja. A **teljes párosítás** olyan párosítás, ami G minden csúcsát lefedi. Az $X \subseteq U$ csúcshalmaz **szomszédság**a az $N(X) \subseteq V$ csúcshalmaz. A páros gráf reprezentálható a szomszédossági mátrixának egy részével is. A sorok az egyik, az oszlopok a másik színosztályt jelentik, az élt 1-es, a nemélt 0 kódolja.

Páros gráfok (ismétlés)

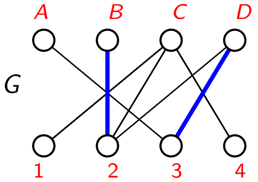


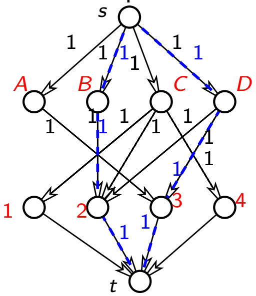
	A	B	C	D
1	0	0	1	0
2	0	$\boxed{1}$	1	1
3	1	0	0	$\boxed{1}$
4	0	0	1	0

Ez itt egy **páros gráf**, **színosztály**ai U és V. **Párosítás** alatt független éleket értünk, amelyeknek nincs közös végpontja. A **teljes párosítás** olyan párosítás, ami G minden csúcsát lefedi. Az $X \subseteq U$ csúcshalmaz **szomszédság**a az $N(X) \subseteq V$ csúcshalmaz.
A páros gráf reprezentálható a szomszédossági mátrixának egy részével is. A sorok az egyik, az oszlopok a másik színosztályt jelentik, az élt 1-es, a nemélt 0 kódolja.
A **párosítás** ebben a reprezentációban **bástyaelhelyezés**nek felel meg: úgy választunk ki néhány 1-est, hogy bmely sorból ill. bmely oszlopból legfeljebb egy 1-es lehet kiválasztva.
A **teljes párosítás** pedig olyan bástyaelhelyezést jelent, amelyben a mátrix minden oszlopában és minden minden sorában áll bástya.

Páros gráfok maximális párosításai (ismétlés)

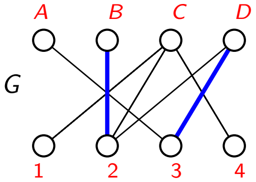
Páros gráf maximális párosítása megfogalmazható maximális folyam feladatként. Irányítsunk minden élt A-ból B-be, vegyünk fel s, t terminálokat, minden sa és minden bt élt, és legyen minden él kapacitása 1. Ha van k ftn él G-ben, akkor a kapott hálózatban van k nagyságú st-folyam. Ha van k nagyságú (egész) st-folyam a hálózatban, akkor G-ben van k méretű párosítás.

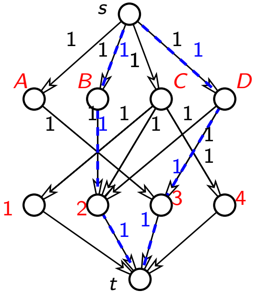




Páros gráfok maximális párosításai (ismétlés)

Páros gráf maximális párosítása megfogalmazható maximális folyam feladatként. Irányítsunk minden élt A-ból B-be, vegyünk fel s, t terminálokat, minden sa és minden bt élt, és legyen minden él kapacitása 1. Ha van k ftn él G-ben, akkor a kapott hálózatban van k nagyságú st-folyam. Ha van k nagyságú (egész) st-folyam a hálózatban, akkor G-ben van k méretű párosítás.





A fenti hálózaton a javító utas algoritmus az EgÉr lemma miatt maximális nagyságú egészfolyamot szolgáltat, így megkapjuk G egy maximális párosítását. Hogy néz ki ez a gyakorlatban?

Páros gráfok maximális párosításai (ismétlés)

A fenti hálózaton a javító utas algoritmus az EgÉr lemma miatt maximális nagyságú egészfolyamot szolgáltat, így megkapjuk G egy maximális párosítását. Hogy néz ki ez a gyakorlatban?

Páros gráfok maximális párosításai (ismétlés)

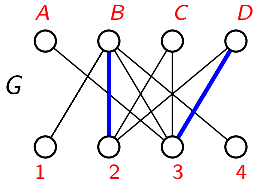
A fenti hálózaton a javító utas algoritmus az EgÉr lemma miatt maximális nagyságú egészfolyamot szolgáltat, így megkapjuk G egy maximális párosítását. Hogy néz ki ez a gyakorlatban? Kiindulunk az $M = \emptyset$ párosításból, és ezt növeljük minden lépésben. Így néz ki egy lépés: M éleit B-ből A-be, G többi élét A-ból B-be irányítjuk, majd A-beli, M által fedetlen csúcsból B-beli, M által fedetlen csúcsba keresünk irányított utat.

Páros gráfok maximális párosításai (ismétlés)

A fenti hálózaton a javító utas algoritmus az EgÉr lemma miatt maximális nagyságú egészfolyamot szolgáltat, így megkapjuk G egy maximális párosítását. Hogy néz ki ez a gyakorlatban?
Kiindulunk az $M = \emptyset$ párosításból, és ezt növeljük minden lépésben. Így néz ki egy lépés: M éleit B-ből A-be, G többi élét A-ból B-be irányítjuk, majd A-beli, M által fedetlen csúcsból B-beli, M által fedetlen csúcsba keresünk irányított utat.
I. Ha van ilyen út, akkor ezen út mentén cserélünk: az út M-beli éleit kidobjuk M-ből, az M-en kívülieket bevesszük M-be. (Tkp az út élei megfordítjuk.) Az így kapott párosítás 2-vel több csúcsot fed, tehát 1-gyel több élből áll, mint a korábbi.

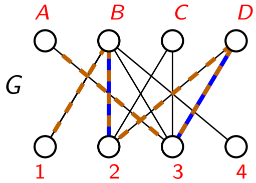
Páros gráfok maximális párosításai (ismétlés)

A fenti hálózaton a javító utas algoritmus az EgÉr lemma miatt maximális nagyságú egészfolyamot szolgáltat, így megkapjuk G egy maximális párosítását. Hogy néz ki ez a gyakorlatban?
Kiindulunk az $M = \emptyset$ párosításból, és ezt növeljük minden lépésben. Így néz ki egy lépés: M éleit B-ből A-be, G többi élét A-ból B-be irányítjuk, majd A-beli, M által fedetlen csúcsból B-beli, M által fedetlen csúcsba keresünk irányított utat.
I. Ha van ilyen út, akkor ezen út mentén cserélünk: az út M-beli éleit kidobjuk M-ből, az M-en kívülieket bevesszük M-be. (Tkp az út élei megfordítjuk.) Az így kapott párosítás 2-vel több csúcsot fed, tehát 1-gyel több élből áll, mint a korábbi.



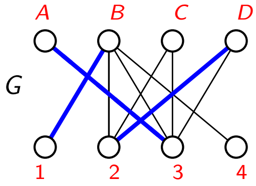
Páros gráfok maximális párosításai (ismétlés)

A fenti hálózaton a javító utas algoritmus az EgÉr lemma miatt maximális nagyságú egészfolyamot szolgáltat, így megkapjuk G egy maximális párosítását. Hogy néz ki ez a gyakorlatban?
Kiindulunk az $M = \emptyset$ párosításból, és ezt növeljük minden lépésben. Így néz ki egy lépés: M éleit B-ből A-be, G többi élét A-ból B-be irányítjuk, majd A-beli, M által fedetlen csúcsból B-beli, M által fedetlen csúcsba keresünk irányított utat.
I. Ha van ilyen út, akkor ezen út mentén cserélünk: az út M-beli éleit kidobjuk M-ből, az M-en kívülieket bevesszük M-be. (Tkp az út élei megfordítjuk.) Az így kapott párosítás 2-vel több csúcsot fed, tehát 1-gyel több élből áll, mint a korábbi.



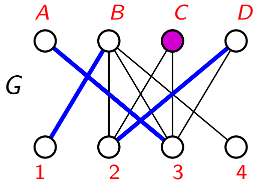
Páros gráfok maximális párosításai (ismétlés)

A fenti hálózaton a javító utas algoritmus az EgÉr lemma miatt maximális nagyságú egészfolyamot szolgáltat, így megkapjuk G egy maximális párosítását. Hogy néz ki ez a gyakorlatban?
Kiindulunk az $M = \emptyset$ párosításból, és ezt növeljük minden lépésben. Így néz ki egy lépés: M éleit B-ből A-be, G többi élét A-ból B-be irányítjuk, majd A-beli, M által fedetlen csúcsból B-beli, M által fedetlen csúcsba keresünk irányított utat.
I. Ha van ilyen út, akkor ezen út mentén cserélünk: az út M-beli éleit kidobjuk M-ből, az M-en kívülieket bevesszük M-be. (Tkp az út élei megfordítjuk.) Az így kapott párosítás 2-vel több csúcsot fed, tehát 1-gyel több élből áll, mint a korábbi.



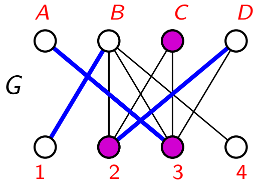
Páros gráfok maximális párosításai (ismétlés)

A fenti hálózaton a javító utas algoritmus az EgÉr lemma miatt maximális nagyságú egészfolyamot szolgáltat, így megkapjuk G egy maximális párosítását. Hogy néz ki ez a gyakorlatban?
Kiindulunk az $M = \emptyset$ párosításból, és ezt növeljük minden lépésben. Így néz ki egy lépés: M éleit B-ből A-be, G többi élét A-ból B-be irányítjuk, majd A-beli, M által fedetlen csúcsból B-beli, M által fedetlen csúcsba keresünk irányított utat.
I. Ha van ilyen út, akkor ezen út mentén cserélünk: az út M-beli éleit kidobjuk M-ből, az M-en kívülieket bevesszük M-be. (Tkp az út élei megfordítjuk.) Az így kapott párosítás 2-vel több csúcsot fed, tehát 1-gyel több élből áll, mint a korábbi.



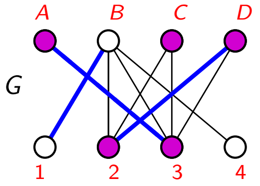
Páros gráfok maximális párosításai (ismétlés)

A fenti hálózaton a javító utas algoritmus az EgÉr lemma miatt maximális nagyságú egészfolyamot szolgáltat, így megkapjuk G egy maximális párosítását. Hogy néz ki ez a gyakorlatban?
Kiindulunk az $M = \emptyset$ párosításból, és ezt növeljük minden lépésben. Így néz ki egy lépés: M éleit B-ből A-be, G többi élét A-ból B-be irányítjuk, majd A-beli, M által fedetlen csúcsból B-beli, M által fedetlen csúcsba keresünk irányított utat.
I. Ha van ilyen út, akkor ezen út mentén cserélünk: az út M-beli éleit kidobjuk M-ből, az M-en kívülieket bevesszük M-be. (Tkp az út élei megfordítjuk.) Az így kapott párosítás 2-vel több csúcsot fed, tehát 1-gyel több élből áll, mint a korábbi.



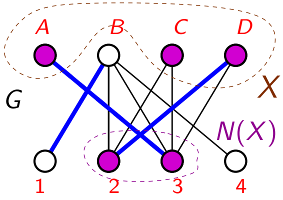
Páros gráfok maximális párosításai (ismétlés)

A fenti hálózaton a javító utas algoritmus az EgÉr lemma miatt maximális nagyságú egészfolyamot szolgáltat, így megkapjuk G egy maximális párosítását. Hogy néz ki ez a gyakorlatban?
Kiindulunk az $M = \emptyset$ párosításból, és ezt növeljük minden lépésben. Így néz ki egy lépés: M éleit B-ből A-be, G többi élét A-ból B-be irányítjuk, majd A-beli, M által fedetlen csúcsból B-beli, M által fedetlen csúcsba keresünk irányított utat.
I. Ha van ilyen út, akkor ezen út mentén cserélünk: az út M-beli éleit kidobjuk M-ből, az M-en kívülieket bevesszük M-be. (Tkp az út élei megfordítjuk.) Az így kapott párosítás 2-vel több csúcsot fed, tehát 1-gyel több élből áll, mint a korábbi.



Páros gráfok maximális párosításai (ismétlés)

A fenti hálózaton a javító utas algoritmus az EgÉr lemma miatt maximális nagyságú egészfolyamot szolgáltat, így megkapjuk G egy maximális párosítását. Hogy néz ki ez a gyakorlatban?
Kiindulunk az $M = \emptyset$ párosításból, és ezt növeljük minden lépésben. Így néz ki egy lépés: M éleit B-ből A-be, G többi élét A-ból B-be irányítjuk, majd A-beli, M által fedetlen csúcsból B-beli, M által fedetlen csúcsba keresünk irányított utat.
I. Ha van ilyen út, akkor ezen út mentén cserélünk: az út M-beli éleit kidobjuk M-ből, az M-en kívülieket bevesszük M-be. (Tkp az út élei megfordítjuk.) Az így kapott párosítás 2-vel több csúcsot fed, tehát 1-gyel több élből áll, mint a korábbi.



Páros gráfok maximális párosításai (ismétlés)

A fenti hálózaton a javító utas algoritmus az EgÉr lemma miatt maximális nagyságú egészfolyamot szolgáltat, így megkapjuk G egy maximális párosítását. Hogy néz ki ez a gyakorlatban?
Kiindulunk az $M = \emptyset$ párosításból, és ezt növeljük minden lépésben. Így néz ki egy lépés: M éleit B-ből A-be, G többi élét A-ból B-be irányítjuk, majd A-beli, M által fedetlen csúcsból B-beli, M által fedetlen csúcsba keresünk irányított utat.
I. Ha van ilyen út, akkor ezen út mentén cserélünk: az út M-beli éleit kidobjuk M-ből, az M-en kívülieket bevesszük M-be. (Tkp az út élei megfordítjuk.) Az így kapott párosítás 2-vel több csúcsot fed, tehát 1-gyel több élből áll, mint a korábbi.

Páros gráfok maximális párosításai (ismétlés)

A fenti hálózaton a javító utas algoritmus az EgÉr lemma miatt maximális nagyságú egészfolyamot szolgáltat, így megkapjuk G egy maximális párosítását. Hogy néz ki ez a gyakorlatban?
Kiindulunk az $M = \emptyset$ párosításból, és ezt növeljük minden lépésben. Így néz ki egy lépés: M éleit B-ből A-be, G többi élét A-ból B-be irányítjuk, majd A-beli, M által fedetlen csúcsból B-beli, M által fedetlen csúcsba keresünk irányított utat.
I. Ha van ilyen út, akkor ezen út mentén cserélünk: az út M-beli éleit kidobjuk M-ből, az M-en kívülieket bevesszük M-be. (Tkp az út élei megfordítjuk.) Az így kapott párosítás 2-vel több csúcsot fed, tehát 1-gyel több élből áll, mint a korábbi.
II. Ha nincs ilyen út, akkor legyen Z az M által fedetlen A-beli csúcsokból irányított úton elérhető csúcsok halmaza és $X := A \cap Z$. Ekkor $N(X) = B \cap Z$, ezért G tetszőleges párosítása az A színosztálynak legalább $|X| - |N(X)|$ db csúcsát fedetlenül hagyja. Az aktuális M párosítás éppen ilyen, ezért az alternáló utas algoritmus maximális párosítást talált, végeztünk.

Alternáló utas algoritmus a mátrixreprezentáción

A páros gráfot reprezentáló 0/1 mátrix 1-eseiből szeretnénk a lehető legnagyobb bástyaelhelyezést kiválasztani. Fedetlen oszlopból szeretnénk fedetlen sorba eljutni úgy, hogy felváltva lépünk vízszinetesn és függőlegesen úgy, hogy ki nem választott és kiválasztott 1-eseken lépkedünk, ki nem választotról indulva és ilyenre érkezve.

Alternáló utas algoritmus a mátrixreprezentáción

A páros gráfot reprezentáló 0/1 mátrix 1-eseiből szeretnénk a lehető legnagyobb bástyaelhelyezést kiválasztani. Fedetlen oszlopból szeretnénk fedetlen sorba eljutni úgy, hogy felváltva lépünk vízszinetesn és függőlegesen úgy, hogy ki nem választott és kiválasztott 1-eseken lépkedünk, ki nem választotról indulva és ilyenre érkezve.

I. Ha van ilyen út, akkor annak a mentén a kiválasztottságot megfordítjuk, ezzel növelve a bástyák számát.

Alternáló utas algoritmus a mátrixreprezentáción

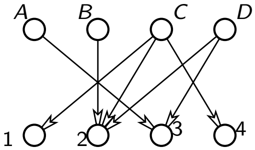
A páros gráfot reprezentáló 0/1 mátrix 1-eseiből szeretnénk a lehető legnagyobb bástyaelhelyezést kiválasztani. Fedetlen oszlopból szeretnénk fedetlen sorba eljutni úgy, hogy felváltva lépünk vízszinetesn és függőlegesen úgy, hogy ki nem választott és kiválasztott 1-eseken lépkedünk, ki nem választotról indulva és ilyenre érkezve.

I. Ha van ilyen út, akkor annak a mentén a kiválasztottságot megfordítjuk, ezzel növelve a bástyák számát.

II. Ha nincs, akkor az ilyen út mentén elérhetó oszlopok halmaza legyen X. Az X-beli oszlopokban álló 1-esek által meghatározott sorok halmaza legyen Y. Világos egyrészt, hogy az X-beli oszlopokban álló bástyák csak Y-beli sorban állhatnak, másrészt pedig $|X| - |Y|$ egyenlő a bástyát nem tartalmazó oszlopok számával. Ezért bármely bástyaelhelyezésben legalább $|X| - |Y|$ db oszlop nem tud bástyát tartlamazni. Az algoritmus épp ilyen bástyaelhelyezéssel állt meg, ezért az algortimus által megtalált bástyaelhelyezés valóban optimális.

Konkrét mátrixon futtatott alternáló utas algoritmus

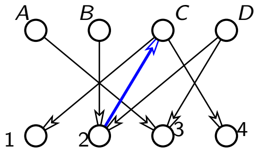
	A	B	C	D
1	0	0	1	0
2	0	1	1	1
3	1	0	0	1
4	0	0	1	0



Bekeretezés a bástyaelhelyezésbe történő kiválasztást, az adott sornál ill. oszlopnál szereplő $\star$ pedig a fedetlen oszlopból induló lépcsős út mentén történő elérhetőséget jelenti.

Konkrét mátrixon futtatott alternáló utas algoritmus

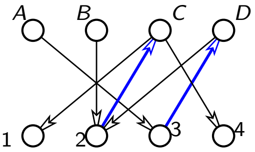
	A	B	C	D
1	0	0	1	0
2	0	1	$\boxed{1}$	1
3	1	0	0	1
4	0	0	1	0



Bekeretezés a bástyaelhelyezésbe történő kiválasztást, az adott sornál ill. oszlopnál szereplő $\star$ pedig a fedetlen oszlopból induló lépcsős út mentén történő elérhetőséget jelenti.

Konkrét mátrixon futtatott alternáló utas algoritmus

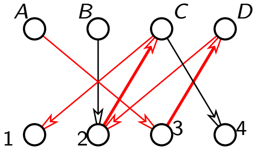
	A	B	C	D
1	0	0	1	0
2	0	1	$\boxed{1}$	1
3	1	0	0	$\boxed{1}$
4	0	0	1	0



Bekeretezés a bástyaelhelyezésbe történő kiválasztást, az adott sornál ill. oszlopnál szereplő $\star$ pedig a fedetlen oszlopból induló lépcsős út mentén történő elérhetőséget jelenti.

Konkrét mátrixon futtatott alternáló utas algoritmus

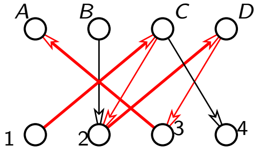
	A	B	C	D
1	0	0	1	0
2	0	1	$\boxed{1}$	1
3	1	0	0	$\boxed{1}$
4	0	0	1	0



Bekeretezés a bástyaelhelyezésbe történő kiválasztást, az adott sornál ill. oszlopnál szereplő $\star$ pedig a fedetlen oszlopból induló lépcsős út mentén történő elérhetőséget jelenti.

Konkrét mátrixon futtatott alternáló utas algoritmus

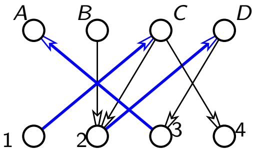
	A	B	C	D
1	0	0	$\boxed{1}$	0
2	0	1	1	$\boxed{1}$
3	$\boxed{1}$	0	0	1
4	0	0	1	0



Bekeretezés a bástyaelhelyezésbe történő kiválasztást, az adott sornál ill. oszlopnál szereplő $\star$ pedig a fedetlen oszlopból induló lépcsős út mentén történő elérhetőséget jelenti.

Konkrét mátrixon futtatott alternáló utas algoritmus

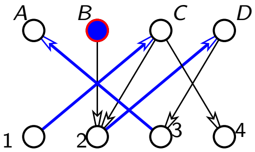
	A	B	C	D
1	0	0	$\boxed{1}$	0
2	0	1	1	$\boxed{1}$
3	$\boxed{1}$	0	0	1
4	0	0	1	0



Bekeretezés a bástyaelhelyezésbe történő kiválasztást, az adott sornál ill. oszlopnál szereplő $\star$ pedig a fedetlen oszlopból induló lépcsős út mentén történő elérhetőséget jelenti.

Konkrét mátrixon futtatott alternáló utas algoritmus

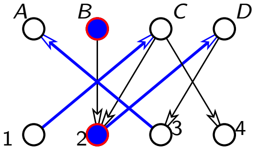
$$\begin{array}{c|cccc|}
 & & \star & & \\
 & A & B & C & D \\
\hline
1 & 0 & 0 & \boxed{1} & 0 \\
2 & 0 & 1 & 1 & \boxed{1} \\
3 & \boxed{1} & 0 & 0 & 1 \\
4 & 0 & 0 & 1 & 0 \\
\hline
\end{array}$$



Bekeretezés a bástyaelhelyezésbe történő kiválasztást, az adott sornál ill. oszlopnál szereplő $\star$ pedig a fedetlen oszlopból induló lépcsős út mentén történő elérhetőséget jelenti.

Konkrét mátrixon futtatott alternáló utas algoritmus

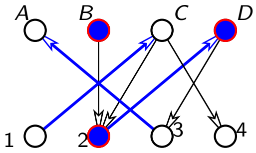
			⋆		
		A	B	C	D
	1	0	0	$\boxed{1}$	0
⋆	2	0	1	1	$\boxed{1}$
	3	$\boxed{1}$	0	0	1
	4	0	0	1	0



Bekeretezés a bástyaelhelyezésbe történő kiválasztást, az adott sornál ill. oszlopnál szereplő ⋆ pedig a fedetlen oszlopból induló lépcsős út mentén történő elérhetőséget jelenti.

Konkrét mátrixon futtatott alternáló utas algoritmus

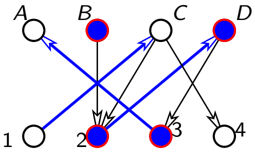
$$
\begin{array}{cc|cccc|}
 & & & \star & & \color{red}{\star} \\
 & & A & B & C & D \\
\hline
 & 1 & 0 & 0 & \boxed{1} & 0 \\
\star & 2 & 0 & 1 & 1 & \color{red}{\boxed{1}} \\
 & 3 & \boxed{1} & 0 & 0 & 1 \\
 & 4 & 0 & 0 & 1 & 0 \\
\hline
\end{array}
$$



Bekeretezés a bástyaelhelyezésbe történő kiválasztást, az adott sornál ill. oszlopnál szereplő $\star$ pedig a fedetlen oszlopból induló lépcsős út mentén történő elérhetőséget jelenti.

Konkrét mátrixon futtatott alternáló utas algoritmus

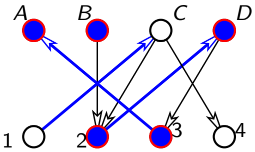
$$
\begin{array}{cc|cccc|}
 & & & \star & & \star \\
 & & A & B & C & D \\
 & 1 & 0 & 0 & \boxed{1} & 0 \\
\star & 2 & 0 & 1 & 1 & \boxed{1} \\
\star & 3 & \boxed{1} & 0 & 0 & 1 \\
 & 4 & 0 & 0 & 1 & 0
\end{array}
$$



Bekeretezés a bástyaelhelyezésbe történő kiválasztást, az adott sornál ill. oszlopnál szereplő $\star$ pedig a fedetlen oszlopból induló lépcsős út mentén történő elérhetőséget jelenti.

Konkrét mátrixon futtatott alternáló utas algoritmus

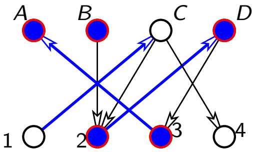
$$\begin{array}{cc|cccc|}
 & & \star & \star & & \star \\
 & & \mathrm{A} & \mathrm{B} & \mathrm{C} & \mathrm{D} \\
\hline
 & 1 & 0 & 0 & \boxed{1} & 0 \\
\star & 2 & 0 & 1 & 1 & \boxed{1} \\
\star & 3 & \boxed{1} & 0 & 0 & 1 \\
 & 4 & 0 & 0 & 1 & 0 \\
\hline
\end{array}$$



Bekeretezés a bástyaelhelyezésbe történő kiválasztást, az adott sornál ill. oszlopnál szereplő $\star$ pedig a fedetlen oszlopból induló lépcsős út mentén történő elérhetőséget jelenti.

Konkrét mátrixon futtatott alternáló utas algoritmus

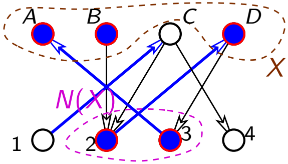
$$\begin{array}{cc|cccc|}
 & & \star & \star & & \star \\
 & & \mathrm{A} & \mathrm{B} & \mathrm{C} & \mathrm{D} \\
\hline
 & 1 & 0 & 0 & \boxed{1} & 0 \\
\star & 2 & 0 & 1 & 1 & \boxed{1} \\
\star & 3 & \boxed{1} & 0 & 0 & 1 \\
 & 4 & 0 & 0 & 1 & 0 \\
\hline
\end{array}$$



Bekeretezés a bástyaelhelyezésbe történő kiválasztást, az adott sornál ill. oszlopnál szereplő $\star$ pedig a fedetlen oszlopból induló lépcsős út mentén történő elérhetőséget jelenti.

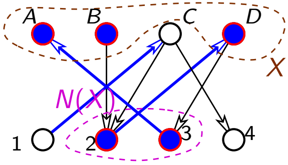
Konkrét mátrixon futtatott alternáló utas algoritmus

$$
\begin{array}{cc||c|c|c|c||}
 & & X & X & & X \\
 & & A & B & C & D \\ \hline
 & 1 & 0 & 0 & \boxed{1} & 0 \\ \hline
Y & 2 & 0 & 1 & 1 & \boxed{1} \\ \hline
Y & 3 & \boxed{1} & 0 & 0 & 1 \\ \hline
 & 4 & 0 & 0 & 1 & 0 \\ \hline
\end{array}
$$



Konkrét mátrixon futtatott alternáló utas algoritmus

		X	X		X
		A	B	C	D
	1	0	0	$\boxed{1}$	0
Y	2	0	1	1	$\boxed{1}$
Y	3	$\boxed{1}$	0	0	1
	4	0	0	1	0



Három méretű bástyaelhelyezést (párosítást) kaptunk. Mivel az X-beli oszlopokban álló 1-esek a két Y-beli sorban helyezkednek el, ezért legfeljebb két X-beli oszlopban állhat bástya. Tehát minden bástyaelhelyezés legalább egy X-beli oszlopot kihagy. Mivel az algoritmus outputja csupán egy X-beli oszlopot hagy ki, ezért a kapott bástyaelhelyezés valóban maximális.

Vége!