1. tétel

 Az optimális hozzárendelés problémája, Egerváry algoritmusa.

## Optimális hozzárendelés

 páros gráf, élsúly-függvény. Keressük maximális összsúlyú *M* teljes párosítást: .

## Címkézés definíció

 páros gráf, címkézés (csúcs súlyfüggény), ha minden élre .

## Lemma

A maximális összsúlyú teljes párosítás összsúlya legfeljebb .

### Bizonyítás

Legyen *M* egy tetszőleges teljes párosítás. Mivel *M* élei minden pontot egyszer fognak le, ezért

## Lemma

 páros gráfban *w* élsúly-függvény, *c* címkézés, *M* teljes párosítás és minden élre teljesül, akkor *M* maximális összsúlyú.

## Egerváry algoritmus

1. lépés: kezdeti címkézés. Legyen ; esetén *c(v)* a *v*-ből induló maximális élsúly, esetén .

2. lépés: *M*-ből kiindulva keresünk egy maximális elemszámú *M*’ párosítást javító utakkal a piros (teljesül rá a ) részgráfban. Ha *M*’ teljes párosítás, akkor STOP (*M’* a keresett párosítás, *c* a keresett címkézés).

3. lépés: Javítóutas algoritmus: *M’* élszáma növekszik vagy címkézés összege csökken. Legyen *F*1 az *M’* áltel le nem fedett *A*-beli pontok halmaza, *L*2 a *B* azon része, melyek az *F*1-ből alternáló úton elérhetők és pirosak (teljesül rá a ). *F*2 az eddigi párosítás szerint *L*2 párja.

Ekkor , majd az új címkézés:

Folytatjuk 2-től (*M*’ és *c*’-vel).

2. tétel

A lineáris programozás alapfeladata, kétváltozós feladat grafikus megoldása. Lineáris egyenlőtlenségrendszer megoldása Fourier-Motzkin eliminációval.

## LP alapfeladat

*A* m×n-es mátrix, *x* egy *n* dimenziós, *b* egy *m* dimenziós, *c* egy *n* dimenziós vektor. LP ekkor:

### Megjegyzések:

* helyett
* helyett és
* helyett

## Kétváltozós

* Minden zárt félsíkot határoz meg, határolója egyenes.
* Ha a félsíkok metszete véges, akkor konvex sokszög.

## Fourier-Motzkin

Az *n* változós esetet visszavezetjük (*n* – 1) változósra. Minden lépésben olyan alakra hozzuk a mátrixot, hogy 1, 0 vagy -1 legyen az első oszlopban.

Ha *A*– üres, akkor *A*+ sorai teljesülnek. *A*+ sorai ilyen alakúak:

Ekkor *x*1-et olyan kicsire választhatjuk, hogy az összes ilyen sort kielégítse. *A*+ sorai így elhagyhatók, elég A0 sorait tovább vizsgálni a 0 oszlop elhagyásával ((*n* – 1) változós eset).

Hasonlóan, ha *A*+ üres, akkor *A*– sorai teljesülnek alapján.

Ha sem *A*+, sem *A*– nem üres, akkor *A*+ és *A*– összes {*i*, *j*} sorpárját összeadva az , egyenlőtlenségrendszer megoldható (az első oszlopra 0 adódik, így (*n* – 1) változós esetre áttérhetünk).

A végén egy változó marad. Ekkor ha *b*0-ban van negatív, akkor a rendszer nem megoldható, mert: sosem teljesülhet, egyébként *A*0 sorai teljesülnek (elhagyhatók). Ha *A*+ vagy *A*– üres, akkor a rendszer megoldható (lásd előző bekezdés). Egyébként megoldható, ha esetén:

 és .

Azaz

3. tétel

Farkas-lemma (két alakban). A lineáris program célfüggvénye felülről korlátosságának feltételei.

## Farkas lemma 1.

Pontosan az egyiknek van megoldása:

### Bizonyítás

Kettőnek egyszerre nincs megoldása, mert: .

Be kell látni még, hogy ha (1) nem megoldható, akkor (2) igen, teljes indukció. Egyváltozós esetben (1) nem megoldható (lásd Fourier-Motzkin egyváltozós eset), ha *b*0-ban van negatív, illetve van egy alakú sorpárja, hogy . Ha *y* eze(ke)n a komponens(ek)ben 1, a többin 0, akkor megoldja (2)-t:

, illetve

vagyis létezik megoldás.

*n*-változós eset: ha *y* megoldás, akkor *α*y is (), ezért elérhető. A rendszer tömören felírva: . Más szóval sorainak *y* szerinti (nemnegatív együtthatós) lineáris kombinációja a vektort adja. Jelölje az eliminációs lépés után -t. Az indukciós feltevés miatt megoldható. Egészítsük ki a rendszert az eliminációs lépés végén elhagyott csupa 0 oszloppal: . Ettől továbbra is megoldható marad, kifejezhető vektor pozitív együtthatós lineáris kombinációja. Visszafelé gondolkodva az eliminációs lépésben egy sora vagy egy sora volt (A0 csoportból), vagy két sorának összege (A+ és A- csoportokból), vagyis sorainak nemnegatív együtthatós lineáris kombinációja kiadja vektort.

## Farkas lemma 2.

Pontosan az egyiknek van megoldása:

### Bizonyítás

Kettőnek egyszerre nincs megoldása:

Ha (2) nem megoldható, akkor (1) igen. (1)-et átírjuk egyenlőtlenségrendszerre, amire alkalmazható a Farkas lemma 1.

## Célfüggvény korlátosság

 megoldható, ekkor az állítások ekvivalensek:

1. Az megoldáshalmazás *cx* felülről korlátos.
2. Nincs megoldása az rendszernek
3. Van megoldása az rendszernek.

### Bizonyítás

(1)🡪(2) indirekt. Legyen *x*0 megoldása -nek, és mégis létezik *z*, hogy: . Ekkor tetszőleges is megoldás, mert . Továbbá nem lehet korlátos felülről, mert és *λ* tetszőlegesen nagy lehet.

(2)🡪(3). Farkas lemma 2-ből adódik ((2)=(2), (3)=(1) helyettesítés), (2)-nél a *z* helyett (*–z*)-t vesszük

(3)🡪(1). , minden *x, y*-ra *yb* felső korlát *cx* értékére.

4. tétel

A lineáris programozás dualitástétele (két alakban). A lineáris programozás alapfeladatának bonyolultsága (biz. nélkül). Egészértékű programozás: a feladat bonyolultsága, korlátozás és szétválasztás (Branch and Bound).

## Dualitás 1.

Ha primál program megoldható és felülről korlátos, akkor

1. duális program is megoldható és alulról korlátos
2. a primál programnak létezik maximuma, a duálisnak minimuma
3. maximum = minimum

### Bizonyítás

(1) bizonyítása a korlátosságnál megvolt, (2) és (3) következik ebből az állításból:

**Állítás**: Tegyük fel, hogy megoldható és *t* tetszőleges. Ha -nek nincs olyan megoldása, amelyre teljesülne, akkor -nak van megoldása, amelyre teljesül.

**Bizonyítás**: Átírva: rendszer nem megoldható, alkalmazzuk rá a Farkas lemma 1-et. Az abban szereplő (2) rendszer *y* vektor utolsó komponensét (ami a hozzáadott egyenlőtlenség) válasszuk külön, jelölje *λ*. Ennek megfelelően kifejtve a lemma (2) egyenleteit:

Ha lenne, akkor teljesülne, így a Farkas lemma alapján nem megoldható, ez ellentmond az állításnak. Ezért , így bevezethetjük . Erre , , teljesül, tehát *y'* teljesíti az állítást. ∎

(2) primál indirekt: nem létezik maximum, de minden felülről korlátos halmaznak van szuprémuma, jelöljük: . Mivel *t* nem maximum, ezért -nek nincs -t teljesítő megoldása. Az előbbi állítás miatt rendszernek van -t teljesítő megoldása. Ekkor: . Itt *yb* *t*-nél kisebb felső korlát *cx*-re, ami ellentmondás, mert *t* a szuprémum (legkisebb felső korlát).

(2) duális: egyszerűen átírjuk: alakra.

(3) fennáll. Indirekt: legyen . Indirekció miatt -nek nincs megoldása, a fenti tétel szerint ekkor a duálisnak van megoldása, amire , ami nyilvánvalóan nem lehet, mert *t* minimális.

## Ekvivalens alak

Ha (primál) program megoldható és felülről korlátos, akkor a
 (duális) program is megoldható és alulról korlátos; a primál programnak létezik maximuma, a duálisnak minimuma, és ezek megegyeznek.

## LP bonyolultsága

* -t kiegyenlítő *x* megoldások közül van-e, ami felülről korlátos *cx*-en?
	+ NP-beli: *x* tanú
	+ coNP-beli: duális megoldása tanú
* szimplex módszer nem polinomiális, de gyors
* ellipszoid módszer polinomiális, de lassú

## Egészértékű programozás

* IP alapfeladat:
* IP duálisa:

## Egészértékű bonyolultsága

* -t kiegyenlítő *x* egész megoldások közül van-e, ami felülről korlátos *cx*-en?
	+ NP-beli: *x* tanú
	+ nincs dualitástétel: nem adódik a co-NP-beli 🡪 NP-teljes

## Branch and Bound

Feladat: , *f* és *g* tetszőleges egész vektorok.

### Jelölések:

* részfeladatok
* *w(i) IP(i)* maximumértéke ennél nagyobb nem lehet
* *z\**: eddigi legjobb célfüggvényérték
* *x\** eddigi legjobb megoldás ()

### Algoritmus

1. lépés: eredeti feladat,
2. lépés: Ha üres, akkor STOP (megoldás *z\**, *x\** helyen). Egyébként vegyünk egy feladatot -ből (és töröljük onnan)
3. lépés: Ha : IP*(i)* nem lehet megoldás, GOTO 2
4. lépés: IP*(i)* helyett LP*(i)* relaxált feladat megoldása. Ha nincs megoldás, akkor GOTO 2. Egyébként maximum *z(i)*, maximumhely *x(i)*
5. lépés
	1. Ha , akkor GOTO 2
	2. Ha , *x(i)* egész vektor. Ekkor z\* és x\* fölülírjuk ezeket az értékekkel. GOTO 2
	3. Ha , *x(i)* nem egész vektor. Ekkor válasszunk *x(i)*-ben egy elágazási változót: xj, és egy közbülső értéket: . Két új feladatot hozunk létre, módosítjuk a korlátokat a megadott komponensre, egyikben: , másikban . A két új célfüggvény felső korlát: . Adjuk hozzá -hez a két új feladatot. GOTO 2.

## Állítás:

Az algoritmus véges sok lépésben leáll és megtalálja a feladat optimumát.

### Bizonyítás

Véges: *f* és *g* miatt, mert csak véges sok egész *x* lehet úgy, hogy .

Optimális indirekt: Legyen *z0* az optimum, de az eljárás csak -t találta. -ben mindig van feladat, aminek az optimuma *z0*. Ez a legelején (0. lépésnél) fönnáll. Ilyenkor 4b vagy 4c lépéshez jut az algoritmus, mert még nincs optimális megoldás. 4b-nél meg is találja 🡪 ellentmondás. 4c esetén az egyik alfeladatban benne van az optimumhely és így optimumérték (*z0*) is. Így viszont sosem ürülne ki, és az algoritmus sosem állna le, ellentmond az előző bekezdéssel.

### Gyakorlati

* LIFO alapján válasszunk új feladatot -ből
* elágazásnál a legkevésbé egész *xj*-t válasszuk elágazási változónak, közbülső értéknek pedig ennek egészrészét.

5. tétel

Bázismegoldások, Caratheodory tétele

## Bázismegoldás definíció

Legyen *x* egy megoldása rendszernek, az *A* mátrix azon sorai, ahol *x* egyenlőséggel teljesíti a feltételt. Ha , akkor *x* bázismegoldás. Ha *x* bázismegoldás és nemnulla komponenseinek megfelelő *A*-beli oszlopok lineárisan függetlenek, akkor erős bázismegoldás.

## Lemma

 rendszer. *A*-ból választunk egy méretű nemszinguláris (det ≠ 0) részmátrixot: *A\**, ehhez tartozó *b* értékeit jelölje *b\**. Az egyenletrendszer egyértelműen megoldható, megoldását (*x\**) egészítsük ki az *A\**-hoz nem tartozó helyeken 0-val. Ha az így kapott *x* megoldja feladatot, akkor erős bázismegoldás. Fordítva: minden erős bázismegoldás előáll ilyen alakban.

### Bizonyítás

Tegyük fel, hogy egy *x* megoldás ilyen alakú. Ekkor *x* egyenlőséggel teljesít *r(A)* darab lineárisan független sort (azokat, amiket *A\**-ba választottunk), tehát , vagyis a bázismegoldást beláttuk. Továbbá a nemnulla komponenseknek megfelelő oszlopok *A\**-hoz tartozó oszlopok közül kerülnek ki, így ezek is függetlenek, tehát az erős bázismegoldás feltétele is teljesül.

Megjegyzés: *A\** lineáris függetlensége a nemszingularitás következménye.

Fordítva: legyen *x* erős bázismegoldás. Válasszunk ki sorai közül tetszőleges *r(A)* darab független sort. Erős bázismegoldás definíciója miatt *x* nemnulla komponenseinek megfelelő *A*-beli oszlopok lineárisan függetlenek, ebből legfeljebb *r(A)* darab van. (Egyébként egészítsük ki tetszőleges oszloppal, hogy *r(A)* darab lineárisan független oszlop legyen.) Adott *r(A)* független sor és *r(A)* független oszlop, ezek kereszteződéseiben kialakuló méretű részmátrix nemszinguláris (lásd BSZ1), illetve a hozzá tartozó *b\** komponens, amelyre: .

## Caratheodory tétel

 rendszer megoldható, megoldáshalmazán *cx* felülről korlátos. Ekkor minden *x0* megoldáshoz létezik olyan *x1* erős bázismegoldás, amelyre: .

### Bizonyítás

Három állítás segítségével látjuk be.

**Állítás1**: Legyen *x0* megoldása -nek. Ha *x0* nem bázismegoldás, akkor létezik olyan *z* vektor, amelyre: , valamint *A*-nak van olyan *ai* sora, amelyre .

**Bizonyítás1**: Ha *x0* nem bázismegoldás, akkor , így *A*-nak van olyan *aj* sora, ami nem fejezhető ki soraiból lineáris kombinációval. Ekkor tehát nem megoldható az egyenletrendszer. Farkas-lemma3 szerint: létezik olyan *z* vektor, melyre: .

Ha , akkor *z* vagy (*–z*) teljesíti az állítást, mert *z*-ről áttérve (*–z*)-re továbbra is teljesül, de előjelet vált (vagyis az állítás második fele is teljesül).

Ha , akkor *z* vagy (*–z*) közül valamelyikre teljesül . Erre alkalmazva a felülről korlátosságra vonatkozó tétel (hivatkozás!) (2) feltételét, lehetetlen, egyéb esetben rendszer nem lenne felülről korlátos *cx* megoldáshalmazán (eredeti tétel). Vagyis ebben az esetben is van olyan olyan *ai* sora, amelyre teljesül. ∎

**Állítás2**: Legyen *x0* megoldása -nek. Ha *x0* nem bázismegoldás, akkor létezik -nek olyan *x’* megoldása, amely *x0*-nál több *A*-beli sort teljesít egyenlőséggel, és amelyre teljesül.

**Bizonyítás2**: Legyen *z* Állítás1 szerinti vektor és legyen . Belátjuk, hogy alkalmas *λ* választással megfelel az állításnak. Vegyük észre, hogy (lásd Állítás1), vagyis *λ*-tól függetlenül *xλ* egyenlőséggel teljesíti sorait. Ahhoz, hogy *xλ* megoldása legyen -nek kell, hogy teljesüljön *A* minden *ai* sorára (ami nem tartozik -be). Két lehetőség van *ai* sorokra vonatkozóan.

Ha , akkor ez a feltétel minden -ra teljesül.

Ha , akkor -ra teljesül a feltétel.

Legyen . Ekkor *xλ* megoldása a rendszernek, sorait egyenlőséggel teljesíti, plusz még legalább egy sort *λ* definíciója miatt. Másrészt . Így megfelel az állításnak. ∎

**Állítás3**: Legyen *x0* bázismegoldása -nek. Ha *x0* nem erős bázismegoldás, akkor létezik -nek olyan *x’* bázismegoldása, amelynek több 0 komponense van, mint *x0*-nak, és amelyre teljesül.

**Bizonyítás3**: Mivel *x0* nem erős bázismegoldás, ezért a nemnulla komponenseinek megfelelő *A*-beli oszlopok lineárisan összefüggőek (erős bázismegoldás definíció, hivatkozás!). Lineáris függetlenség definíciója szerint létezik olyan vektor, amelyre . Mivel , ezért csak akkor teljesül, ha is teljesül, röviden: .

Tekintsük újra vektort (*λ* tetszőleges). Megmutatjuk, hogy *λ* minden értékére *xλ* bázismegoldás. Ugyanis , ezért *xλ* megoldás, továbbá miatt ugyanazokat a sorokat teljesíti egyenlőséggel, mint *x0*, vagyis , így bázismegoldás is. Emellett is igaz, különben *z* helyett (*–z*)-re áttérve teljesülne, így a felülről korlátosság tétel (2) feltétele szerint *cx* nem volna felülről korlátos megoldáshalmazán. Ezért .

Meg lehet mutatni, hogy alkalmas *λ* választásával *xλ*-nak több 0 komponense lehet, mint *x0*-nak. A feltétel miatt nyilván is teljesül minden *i*-re és *λ*-ra. Válasszunk egy olyan *j* komponenst, hogy ( miatt van ilyen komponens), ekkor is igaz. Válasszuk *λ*-t úgy, hogy teljesüljön, ekkor , ekkor -nak több 0 komponense van. ∎

#### Eredeti bizonyítás folytatása

Induljunk ki egy tetszőleges *x0* megoldásból. Létezik olyan *x1* bázismegoldás, hogy (Állítás1 és Állítás2 miatt). Ha *x1* erős bázismegoldás, akkor megfelel a tételnek. Ha nem, akkor alkalmazzuk rá Állítás3-at. Mivel véges komponense van a vektornak, ezért az eljárás véges sok lépés után leáll. (Legvégső esetben nullvektor adódik, de ez erős bázismegoldás is).

6. tétel

Egészértékű programozás totálisan unimoduláris együtthatómátrixszal. Alkalmazás páros gráfokra.

## Totális unimoduláris (TU) mátrix

Minden négyzetes részmátrixának determinánsa 0, 1 vagy –1. Szükséges (de nem elégséges), hogy a mátrix elemei is csak 0, 1 vagy –1 értékűek lehetnek.

## Lemma

Egy mátrix totális unimoduláris marad, ha

1. egy sorát/oszlopát (–1)-gyel szorozzuk
2. egységvektort hozzáveszünk sorként/oszlopként
3. egyik sorát/oszlopát új sorként/oszlopként hozzávesszük
4. transzponáljuk

### Lemma bizonyítása

Nyilván csak azoknak a négyzetes részmátrixoknak változhat a determinánsa, amelyikeket érint a változás.

1. (1): a determináns (–1)-szeres lesz
2. (2): a determináns megegyezik vagy (–1)-szeres lesz az egység pozíciójától függően.
3. (3): ha a régi és új sor/oszlop is szerepel a kiválasztott részmátrixnak, akkor a determináns 0. Ha csak az egyik (vagy a régi vagy az új), akkor előáll az eredeti mátrixból képzett részmátrix, determináns marad 0, 1 vagy –1.
4. determináns definíciójának következménye

## Egészértékű totálisan unimoduláris együtthatómátrixszal

Legyen *A* totálisan unimoduláris mátrix, *b* egész koordinátájú vektor. A (LP) feladat megoldható és maximuma véges. Ekkor a (IP) feladat is megoldható, és maximuma megegyezik (LP) feladat maximumával.

### Bizonyítás

Cramer-szabály segítségével.

#### Cramer-szabály

Legyen *A* egy -es, nemszinguláris mátrix, *b* egy *n*-dimenziós vektor. Jelölje *Ai* azt a mátrixot, amelyet *A*-ból úgy kapunk, hogy az *i*-edik oszlopát helyettesítjük *b* vektorral. Ekkor az rendszer egyértelmű megoldásának *i*-edik komponense: .

#### Bizonyítás folytatása

Azt kell belátnunk, hogy rendszer minden erős bázismegoldása egész vektor, hiszen Caratheodory tétel alapján a maximum felvétetik erős bázismegoldáson is.

A Lemma alapján az erős bázismegoldások előállnak úgy, hogy *A*-ból kiválasztunk egy méretű, nemszinguláris *A\** részmátrixot, megoldjuk a megfelelő egyenletrendszert, és *x\**-ot kiegészítjük 0 komponensekkel. Cramer-szabály szerint *x\** minden komponense egy tört, számlálója *A\** és *b\** elemeiből képzett determináns, nevezője minden esetben det(*A\**). Ekkor a számláló mindig egész (*A* és *b* is egész értékeket tartalmaz, a szorzatok összege is egész), továbbá az unimodularitás miatt (nemszinguláris, tehát 0 nem lehet).

## Tétel

Minden irányított gráf illeszkedési mátrixa totálisan unimoduláris.

### Bizonyítás

Teljes indukció: *M* egy méretű részmátrix. Ha , akkor nyilvánvaló, mert minden elem 0 vagy 1.

Ha és *M*-nek van olyan oszlopa, amelyben legfeljebb egy nemnulla elem van (azaz egységvektor, vagy annak ellentetje). Ha ezt az oszlopot elhagyjuk, akkor az indukciós lépés miatt a maradék mátrix TU. Ha hozzáveszünk egy egységvektort (lemma (2)) vagy ellentettjét (lemma (1)), TU marad.

Egyébként minden oszlopban egy +1 és egy –1 elem van, ekkor *M* sorainak összege nullvektor, a determináns 0.

## Tétel

Páros gráf illeszkedési mátrixa totálisan unimoduláris.

### Bizonyítás

Felhasználjuk az irányított gráf tételét: páros gráf éleit irányítsuk úgy, hogy minden él *A*-ból *B*-be mutasson. A *B*-hez tartozó sorokat negáljuk, de ez nem változtat TU tulajdonságon (lemma (1)).

## Maximális összsúlyú párosítás IP feladatként

Jelölések: *x* indikátor, hogy egy él benne van-e a párosításban. *w* tetszőleges (él)súlyfüggvény, B illeszkedési mátrix (páros gráf lévén TU)

### Feladat

. Az feltétel miatt *B*-t kiegészítjük az -es egységmátrix ellentettjével, de ez nem változtat TU tulajdonságon. A azt jelenti, hogy az egy csúcsból kiinduló élek közül legfeljebb egyet választunk ki 🡪 párosítás. Mivel *B* TU mátrix és *b* egész vektor, ezért *x* egész vektor (IP feladat) (pontosabban 0 vagy 1 értékű).

### Duális

. Ekkor *y* megoldás minden *v* csúcshoz egy *c(v)* címkét rendel. A azt jelenti, hogy minden élre (Egerváry Jenő).

7. tétel

A lineáris és egészértékű programozás alkalmazása hálózati folyamproblémákra.

## Jelölések

 irányított gráf

* két kitüntetett csúcs
* nemnegatív *kapacitásfüggvény*. tetszőleges függvény.
* *ρx(v)*: *v*-be belépő élek összege *x* szerint
* *δx(v)*: *v*-ből kilépő élek összege *x* szerint
* Egy *x* függvény akkor *folyam*, ha minden esetén
* *x* függvény megengedett, ha minden *e* élre:
* A folyam értéke .

## Lemma

Legyen olyan függvény, amely minden esetén , továbbá . Ekkor *x* folyam.

### Bizonyítás

Minden élből vegyünk fel egy új élt, az új gráf legyen *G’*. Minden új *e* élhez rendeljük hozzá az értéket (amennyivel több a belépő, mint a kilépő), a régi éleken maradjon az érték . A kapott *x’* függvény *G’*-ben nyilván folyam, ezért . Másrészt és . A lemma feltételei alapján: . Mivel csak egyenlőség állhat, ezért , ami csak akkor teljesülhet, ha *x* folyam.

## LP felírás

Egészítsük ki G gráfot egy pszeudó éllel, (értéke lesz majd folyam értéke, jelöljük *µ*-vel), az így kapott gráf illeszkedési mátrix legyen *B\**. Minden csúcshoz tartozik egy sor, melyre teljesül (ez folyamnál azt jelenti, hogy a belépő élek összege nem kisebb, mint a kilépőké; ). rendszer alapján , illetve , összevontan: . Az előző lemma miatt , vagyis a folyam értéke *µ*.

Keressük a maximális folyamot, vagyis . Az feltételt hozzávesszük a mátrixhoz (ez lesz az *E* egységvektor és a hozzá tartozó *c* rész a *b* vektorban) az alábbi formában, ezt jelöljük *M*-mel.

## Duális

A . Írjuk fel másképp *y*-t: . Ekkor:

1. ,
2. minden él esetén

Ekkor a cél minimum.

## Állítás

A értéke megegyezik a hálózati folyam minimális vágásának értékével (*mC*).

### Bizonyítás

Egy *mC* vágáshoz könnyen készíthető olyan *π* és *w*, amelyre , a következők szerint. Legyen *S* () és *T* () diszjunkt halmazok, akkor esetén , esetén legyen . Minden *e* él, ami *S*-ből *T*-be mutat: , egyébként . A teljesül, ebből adódik, a másik irányú egyenlőtlenséget kell belátni.

Az M mátrix TU, ezért (Egészértékű totálisan unimoduláris együtthatómátrixszal) alapján *y* is egészértékű (mivel rendszerbeli *c* is egészértékű). Legyen adott (*π, w*) optimális, egészértékű megoldás, ebből kiindulva elkészítünk egy (*π’, w’*) 0 vagy 1 értékű optimális megoldást. Legyen

Ekkor (*π’, w’*) (1) és (3) nyilván teljesül

(2) indirekt: él esetén , ekkor a 0 – 1 értékűség miatt , esetben valósulna meg, mert *w’* definíciója miatt .

Optimális, mert , mert .

Legyen és a fentiek szerinti *S* és *T*. Minden élre (2) miatt . Minden más élen *w’(e)* csak akkor lehet 1, ha , mert -ra változtatása után a feltételek továbbra is fennállnának, így .

8. tétel

Matroid definíciója, alapfogalmak (bázis, rang, kör). Példák: lineáris matroid (mátrixmatroid), grafikus matroid, uniform matroid. A rangfüggvény szubmodularitása.

## Matroid definíció (függetlenségi axiómák)

Legyen *E* tetszőleges véges halmaz, matroid, ha:

1. és
2. és , akkor létezik olyan , amelyre

## Alapfogalmak

Az matroidban az alaphalmaz -hez tartozó részhalmazait *független* halmazoknak nevezzük. A maximális (nem bővíthető) független halmazok a matroid *bázis*ai. Egy minimálisan összefüggő (egy elem elvételével már független) halmazt *kör*nek nevezzük. Az egyelemű kör *hurok*. Az *M* matroidban egy halmaz *rang*ja *r(X)*: egy *X*-beli maximális független halmaz mérete.

## Lemma

Legyen matroid, . Ha *X1* és *X2* maximális független halmazok *A*-ban, akkor .

### Bizonyítás

Indirekt: . (F3) miatt létezik olyan , amelyre , vagyis *X1* nem lehetett maximális.

## Példák

* *Grafikus matroid* (körmatroid): *G* gráf által indukált matroid. , {*G*-beli erdők}
* *Lineáris matroid*: *A* mátrix által indukált matroid. , {*A* lineárisan független oszlopai}.
* *Uniform matroid*: *E*: tetszőleges véges halmaz, jelölje *n* az elemszámát. {*E* legfeljebb *k*-elemű részhalmazai} (), jelölése: *Un,k*. Speciálisan: *Un,n*: teljes/szabad matroid, *Un,0*: trivális matroid.

## Rangfüggvény szubmodularitás

Legyen *r* egy matroid rangfüggvénye. Ekkor minden halmazpárra:

1. .
2. , ha

Fordítva: ha *r* egy egészértékű függvény *E* részhalmazain, amelyre teljesül (R1) – (R4), akkor *r* egy matroid rangfüggvénye, ahol .

### Bizonyítás

(R1) – (R3) a rangfüggvény definíciójából adódik.

(R4): adott *X, Y* halmazpárra legyen *A* maximálisan független, , elemszáma . Az *A* halmaz kiterjeszthető egy olyan *B* halmazzá, amely független lesz -ban. *X*-ből *β*, *Y*-ból *γ* új elem kerül *B*-be. A rangfüggvények a következőképp alakulnak:

Összesítve:

9. tétel

Mohó algoritmus matroidon. Matroid megadása rangfüggvényével, bázisaival (biz. nélkül). Matroid duálisa, a duális matroid rangfüggvénye.

## Mohó algoritmus

Legyen matroid, nemnegatív súlyfüggvény. Keressük a maximális összsúlyú független halmazt, azaz: . Tetszőleges matroidra és súlyfüggvényre optimális megoldást (maximális összsúlyú) ad a mohó algoritmus.

### Bizonyítás

Indirekt. A mohó algoritmus megoldást adta, de az optimális . Mivel mindkét halmaz maximálisan független, ezért (F3) miatt . Legyen mindkét halmaz költség szerint csökkenő sorrendbe rendezve, vagyis és . Tudjuk, hogy a mohó algoritmus a legnagyobb súlyú elemmel kezd, ezért .

Legyen *i* a legkisebb index, amire Ilyen mindenképp létezik, különben a mohó algoritmus optimális megoldást adott volna. Legyen és . Alkalmazzuk (F3)-at erre:

Létezik , amelyre . Feltettük, hogy csökkenő sorrendbe vannak rendezve a halmazok elemei, vagyis , továbbá miatt az algoritmus *bj*-t választotta volna, nem *ai*-t.

## F3 vs mohó

Legyen matroid, nemnegatív súlyfüggvény. (F1) – (F2) teljesül, továbbá tetszőleges *w* költségfüggvényre a mohó algoritmus maximális súlyú megoldást ad: . Ekkor (F3) is teljesül.

### Bizonyítás

Indirekt: mohó algoritmus OK, de (F3) nem teljesül. Ilyenkor , amelyeknél fennáll, de nem létezik olyan , amelyre teljesülne. Legyen és súlyfüggvény a következő:

Az algoritmus először kiválasztja *Y* elemeit. Ezt követően az indirekt feltevés miatt már nem választhat *X – Y*-beli elemeket, csak 0 súlyúakat. Ekkor az összsúly lesz. Azonban *X* összsúlya ennél nagyobb, tehát a mohó algoritmus nem találta meg az optimális megoldást 🡪 ellentmond a kezdeti feltevésnek.

### Bázisos megadás

Legyen egy matroid bázisainak a halmaza. Ekkor

1. minden -re
2. Ha és , akkor létezik olyan , melyre .

Fordítva: ha egy halmazrendszer, amelyre teljesül (B1) – (B3), akkor matroid, ahol .

## Rangfüggvény megadás

Legyen *r* egy matroid rangfüggvénye. Ekkor minden halmazpárra:

1. .
2. , ha

Fordítva: ha *r* egy egészértékű függvény *E* részhalmazain, amelyre teljesül (R1) – (R4), akkor *r* egy matroid rangfüggvénye, ahol .

## Duális matroid definíció

 matroid bázisai , akkor a duális matroid bázisai . Ebből már adódik

## Duális matroid tétel

Az matroid.

### Bizonyítás

(F1) – (F2) nyilvánvalóan teljesül, (F3)-at kell belátni: ha és , akkor létezik olyan , amelyre . Legyen és bázis az eredeti matroidban. Ha van olyan elem *X*-ben, ami nincs benne *Y*-ban, sem *BY*-ban, akkor ezt *Y*-hoz hozzávéve ismét -beli elemet kapnánk.

Egyébként nem ilyen egyszerű a helyzet, azaz . Ekkor a feltevés miatt. Mivel *BX* és *BY* ugyanakkora, ebből következik. Alkalmazzuk erre a két halmazra (és *M* matroidra) (F3)-at: létezik olyan elem, amelyre független halmaz *M*-ben. Ezt a független halmazt egészítsük ki bázissá úgy, hogy *BY*-ból veszünk hozzá új elemeket, jelöljük *B’*-vel.

*B’* tartalmaz elemet -ból, tehát létezik olyan elem *BY*-ban – jelöljük *u*-val –, ami nincs B’-ben. Ekkor elemre teljesül: miatt .

## Duális rangfüggvény

.

### Bizonyítás

10. tétel

Elhagyás és összehúzás. Matroidok direkt összege, összefüggősége. T test felett reprezentálható matroid duálisának T feletti reprezentálhatósága.

## Elhagyás

 matroid, elhagyása: , ahol .

## Összehúzás

 matroid, összehúzása: , ahol *M/X* rangfüggvénye: .

## Tétel

Az elhagyások és összehúzások fölcserélhetők. Minden *M* matroid *N* minora (elhagyások és összehúzások sorozata) előáll alakban, ahol *A* és *B* diszjunkt halmazok.

## Tétel

Elhagyás és összehúzás duális művelet: és .

### Bizonyítás

Elég az elsőt bizonyítani. A rangfüggvénye:

 rangfüggvénye:

ahol az *M\X* matroid alaphalmaza. Ebből adódik.

## Matroidok direkt összege

Legyen és két matroid a diszjunkt *E*1 és *E*2 nemüres alaphalmazon. A két matroid direkt összege az az matroid, melynek alaphalmaza , és egy halmaz akkor független *N*-ben, ha és független *M1*-ben, illetve *M2*-ben.

## Matroidok összefüggősége

Egy matroid összefüggő, ha nem áll elő matroidok direkt összegeként.

## T test felett reprezentálható

Az matroid reprezentálható *T* test felett, ha *E* minden eleme *T* feletti vektor.

### Definíció2

Az matroid reprezentálható *T* test felett, ha létezik olyan mátrix, amelynek oszlopai *T* feletti vektorok, és ezek által meghatározott lineáris matroid izomorf *M*-mel.

## Duális reprezentációja T test felett

Ha matroid reprezentálható *T* test felett, akkor *M\** is.

### Bizonyítás

Induljunk ki a reprezentálás definíciójából: az alaphalmaz elemszámát jelöljük , *M* rangját *r*-rel, *M*-mel izomorf lineáris matroidot legyen *A*. Ekkor írjuk föl alakban, *Er* az -es egységmátrix (*M* egy bázisa), *A*0 a többi eleme. Belátjuk, hogy reprezentálja *M\**-ot.

Mindkét mátrixnak *n* oszlopa van, így az alaphalmazok megfeleltethetők egymásnak, a rangok is rendben vannak. Válasszuk ki *M* egy *B*1 bázisát (nyilván *r* oszlop), ezt válasszuk úgy, hogy *Er* utolsó *t* darab oszlopa, és *A*0 első *r – t* oszlopa (ábrán a színes). Világos, hogy az egységmátrixból vett oszlopokhoz tartozó oszlopok és sorok (sárgával) determinánsa nem lehet 0, egyedül *C* részmátrix kérdéses. Mivel *B*1 bázis, ezért az oszlopai függetlenek *C* determinánsa sem 0.

A duális matroidban azok az oszlopok alkotnak bázist, amik eredetileg nem, mert . Vagyis az első *r – t* és az utolsó *n – 2r + t* oszlop. Ez esetben sem kérdéses az egységmátrixhoz tartozó sorok és oszlopok függetlenek (determinánsuk ≠ 0), a maradék (narancssárga) rész a kérdéses. Azonban a konstrukció miatt ez pont *C* transzponáltja, tehát determinánsa nem lehet 0, ezért valóban bázis.



11. tétel

Grafikus, kografikus, reguláris, bináris és lineáris matroid fogalma, ezek kapcsolata (ebből bizonyítás csak a grafikus és reguláris matroidok közötti kapcsolatra), példák. Fano-matroid, példa nemlineáris matroidra. Bináris, reguláris és grafikus matroidok jellemzése tiltott minorokkal: Tutte tételei (biz. nélkül). Seymour tétele (biz. nélkül).

## Matroid osztályok

*Grafikus matroid* (körmatroid): *G* gráf által indukált matroid. , {*G*-beli erdők}. Grafikus matroid duálisa *kografikus*.

*Reguláris* matroid, ha minden test felett reprezentálható. A *bináris* matroid a bináris test felett reprezentálható. Ha van olyan test, ami felett reprezentálható, akkor *lineáris* matroid.



## Tétel

Grafikus matroid bármilyen test felett reprezentálható (reguláris).

### Bizonyítás

Rendeljünk a gráf minden pontjához egy-egy különböző *n*-dimenziós egységvektort (n-pontú gráf esetén). Az élekhez pedig a két végpontja közti különbséget (irányítás lényegtelen). Az élek vektorában csak 0, 1 és –1 szerepelhet, így minden test felett reprezentálható.

Ha egy kör éleit ±1 együtthatós lineáris kombinációval összeadjuk, akkor nullvektort kapunk (–1-re azért van szükség, mert nem törődtünk az élek irányításával), mert a megfelelő nemnulla komponensek kiejtik egymást.

Fordítva: vegyünk egy összefüggő vektorhalmazt, ahol egyik együtthatója sem 0, jelölje *X*. Az élvektorok azokon a koordinátákon nemnulla, amelyik pontokat összeköt. Ahhoz, hogy a nullvektor kijöjjön, minden egy ponthoz két nemnulla koordináta kell, azaz egy pontra két él illeszkedik (a foka legalább 2). Ha valamely részgráf, ahol minden pont foka 2 (vagy több), akkor biztos van benne kör.

## Fano matroid

Adott a hételemű halmaz: . A legfeljebb kételemű részhalmazok függetlenek, továbbá a háromeleműek közül azok, amik nincsenek egy egyenesen vagy körön, ez a Fano-matroid (*F*7). A legfeljebb kételemű részhalmazok függetlenek tovább azok a háromeleműek, amik nincsenek egy egyenesen, ez az anti-Fano matroid ().

### Következmény

A Fano-matroid pontosan azon testek felett reprezentálható, amelynek karakterisztikája 2 (pl bináris matroid). Az anti-Fano-matroid pontosan azon testek felett reprezentálható, amelyek karakterisztikája nem 2. Ezért a két matroid direkt összege nem reprezentálható semmilyen test felett 🡪 nem lineáris.

## Tutte tételei

* M matroid *bináris* ⇔ nem tartalmaz minorként *U*4,2 matroidot.
* M matroid *regurális* ⇔ nem tartalmazza minorként *U*4,2, *F*7 és *F*7\* matroidokat.
* M matroid *grafikus* ⇔ nem tartalmazza minorként *U*4,2, *F*7, *F*7\*, *M*\*(*K*5) és *M*\*(*K*3,3) matroidokat.

## Seymour tétel

M reguláris ⇔ előáll 1 grafikus, 1 kografikus és egy R10 matroid néhány példányából a direkt összeg, 2-összeg és 3-összeg műveletek segítségével.

12. tétel

Matroidok összege. k-matroid-metszet probléma, ennek bonyolultsága k ≥ 3 esetén.

## Matroid összege def

 és matroidok összege , ahol , hogy és , illetve . (*X* előáll egy -beli és egy -beli elem uniójaként)

## Tétel

Matroidok összege is matroid.

### Bizonyítás

(F1), (F2) nyilvánvaló, (F3) indirekt: és , akkor nem létezik olyan , amelyre

Definíció szerint *X*, *Y* halmazoknak létezik (esetleg többféle) és felbontása, ahol , illetve . Válasszunk olyat, hogy a részhalmazok diszjunktak legyenek és érték legyen minimális, továbbá legyen . Ekkor *M*1 matroidra alkalmazva (F3)-mat, létezik , amelyre . Ha , akkor teljesülne, ami ellentmond az indirekciós feltevésnek. Tehát . Ekkor egy másik felbontás, ráadásul összeg eggyel kevesebb, mint az mint érték, ami ellentmondás (minimálisra választottuk).

## 2-matroid metszet probléma

k-matroid metszet probléma (MMPk): adott *k* darab matroid közös alaphalmazon: . Létezik-e valamely konstans *p*-re *p*-méretű halmaz -ben.

2-matroid metszet probléma: két matroid esetén,

## MMPk bonyolultsága

MMPk NP-teljes (k≥3) esetén

### Bizonyítás

NP-beli, mert *p*-elemű közös bázis tanú

MMP3 visszavezethető irányított Hamilton-út keresésére, ami NP-teljes.

* M1=(E,F1)-ben X⊆E-re X∈F1 ⇔ az X részgráfban minden pont be-foka legfeljebb 1 és u be-foka 0.
* M2=(E,F2)-ben X⊆E-re X∈F2 ⇔ az X részgráfban minden pont ki-foka legfeljebb 1 és v ki-foka 0.
* M3 a gráf körmatroidja.

M1, M2 és M3 közös |V|–1 elemű bázisai G irányított Hamilton-útjai.

13. tétel

A k-matroid partíciós probléma, ennek algoritmikus megoldása. A 2-matroid-metszet feladat visszavezetése matroid partíciós problémára.

14. tétel

Matroidok megadása, orákulumok, ezek kapcsolata. k-polimatroid rangfüggvény fogalma. A 2-polimatroid-matching probléma, ennek bonyolultsága, Lovász tétele (biz. nélkül).

Közelítő és ütemezési algoritmusok

15. tétel

Additív hibával közelítő algoritmus fogalma, példák. k-approximációs algoritmusok, példák: minimális lefogó ponthalmaz, halmazfedés.

## Additív hiba

Egy algoritmus *C* additív hibával közelítve old meg egy minimalizálási problémát, ha minden *I* inputra polinom időben ad egy megoldást, amire:

Példa: Síkbarajzolható gráfok színezése. 5 színnel minden gráf színezhető polinomiális idő alatt (az optimális 3) 🡪 létezik 2 additív hibával közelítő algoritmus a problémára.

## Multiplikatív hiba

Egy algoritmus multiplikatív hibával közelítve old meg egy minimalizálási problémát, ha minden *I* inputra polinom időben ad egy megoldást, amire:

## Minimális lefogó ponthalmaz

A lefogó pontok minimális számát () határozzuk meg úgy, hogy kiválasztjuk a független élek maximális rendszerét (), akkor mindkét végpontját tekintve -elemű ponthalmazt kapunk. Világos, hogy , ezért multiplikatív hibáról van szó.

## Halmazfedés

Egy *n* elemű *U* alaphalmaz részhalmazai , és költségfüggvény. Cél minimális összköltségű fedést találni *U*-ra.

### Algoritmus

 már lefedett halmazok úniója. Egy új *Si* halmaz értéke legyen a hozzávétele esetén az egy új elemre jutó átlagos költség, vagyis: . Ez alapján a mohó algoritmus -szeres multiplikatív hibával közelíti az optimumot.

16. tétel

Közelítő algoritmus a Steiner-fa problémára, illetve a metrikus utazóügynök problémára (Christofides algoritmusa). Az általános utazóügynök probléma közelíthetősége.

## Steiner-fa probléma

Adott egy összefüggő gráf, élein költségfüggvény. *V* két részre van osztva: *T*-beli terminálokra és *S*-beli Steiner pontokra. Cél minimális költségű fát keresni *G*-ben, ami az összes terminált tartalmazza (és esetleg pár Steiner-pontot is). Ha , akkor a feladat mohó algoritmussal hatékonyan megoldható, különben NP-nehéz.

A metrikus Steiner-fában a költségfüggvény kielégíti a háromszög-egyenlőtlenséget. Ez esetben létezik *k*-approximációs algoritmus.

## Tétel

Az általános Steiner-fa probléma visszavezethető a metrikus Steiner-fa problémára, úgy hogy a *k*-approximáció megmarad.

### Bizonyítás

Legyen *G*’ a teljes gráf *V* ponthalmazon. Egy *G*’-beli {*u*, *v*} él költsége a *G*-beli legkisebb összköltségű *u* – *v* út hossza. Mivel minden {*u*, *v*} él költsége *G*’-ben legfeljebb akkora, mint *G*-ben, ezért az optimális megoldás értéke is legfeljebb akkora, mint *G*-ben.

Legyen *F*’ egy *G*’-beli Steiner-fa. Helyettesítsük *F*’ éleit *G*-ben a legrövidebb utakkal, hogy megkapjuk *F* Steiner-fát. *G*’ és *G* csak élekben különbözik, így *F*’ után *F* is tartalmazza az összes terminált, összköltsége . Mivel *F* polinom időben számítható *F’*-ből, ezért done.

## Tétel

Legyen *F* egy minimális költségű feszítőfa a terminálok halmazán. Ekkor .

### Bizonyítás

Vegyünk egy optimális Steiner-fát. Ennek éleit megduplázva olyan összefüggő gráfot kapunk, ami minden terminált összeköt. Ebben keressünk Euler-kört (van rá hatékony algoritmus). A költsége 2*OPT*.

Vegyük sorba az Euler-kör csúcsait, és dobjuk el azokat, amelyeket nem először érintettünk (mivel *G* teljes gráf, továbbra is létezik út, és a háromszög-egyenlőtlenség miatt az összköltség sem növekszik). A csúcsokat összekötve kaptunk egy Hamilton-utat a *T* halmazon, melynek költsége legfeljebb 2*OPT*. Ez az út feszítőfa is a terminálokon, tehát .

## Utazó ügynök

Adott egy *G* teljes gráf, nemnegatív élsúly függvény. Keressük a minimális összsúlyú Hamilton-kört.

## Christofides: metrikus utazó ügynök

1. *F* minimális összsúlyú feszítőfa *G* teljes gráfban
	1. Vegyük azokat a pontokat, amelyek *F*-beli fokszáma páratlan, jelöljük ezeket a pontokat *H*-val (*H* elemszáma páros)
	2. Keressünk *H*-ban minimális összsúlyú teljes párosítást, legyen *M*. *F* és *M* élei között keresünk Euler-kört.
2. Útlevágásokkal leszűkítjük Hamilton-körré.

## Általános utazó ügynök közelítése

Általános utazó ügynök nem közelíthető *k*-approximációval. A probléma visszavezethető (egy nem teljes gráfban) Hamilton-kör keresésére.

Legyen *G* egy *n*-pontú (nem teljes) gráf, a bemenet, hozzá tartozó *G*’ pedig teljes gráf. Ha *G*-ben szomszédos két él, akkor *G*-ben egységnyi az élsúly, egyébként *kn*. Ha *G*-ben van Hamilton-kör, akkor *G*’-ben a minimális összsúlyú Hamilton-kör összsúlya *n*, egyébként legalább . Mivel ez több, mint az optimum *k*-szorosa, ezért nincs ilyen közelítés.

17. tétel

Polinomiális approximációs séma a Részösszeg problémára

18. tétel

Ütemezési feladatok típusai. Az 1|prec|Cmax és az 1||PCj feladat. 2-közelítő algoritmus a P||Cmax feladatra.

## Ütemezési feladatok osztályozása

Az ütemezési feladatok egy α|β|γ hármassal írhatók le, ahol

* α a gépek száma
	+ 1: 1 gép
	+ Pm: m párhuzamosan futó gép
	+ P: nem rögzített számú párhuzomosan futó gép. P|β|γ az {1|β|γ, P2|β|γ, P3|β|γ, ...} feladatokat tartalmazó osztály neve.
* β az infók halmaza az ütemezésről. Pl.:
	+ prec: adott egy irányított gráf, ami megkötéseket tartalmaz arra nézve, hogy egy adott munka elkezdéséhez mely munkákat kell előbb befejezni
	+ rj: adottak a release time-ok, azaz hogy melyik job mikortól áll rendelkezésre
	+ pj: adottak a megmunkálási idők, mennyit vesz igénybe az adott munka elvégzése
* γ a függvény, ami szerint optimalizálunk
	+ Cmax = max(Cj): az utolsó munka befejezési ideje
	+ ∑Cj ~ ∑Cj/n: a munkák átlagos befejezési ideje

## 1|prec|Cmax

A feladatokat topologikus sorrendben adogatjuk a gépnek. Ez optimális, mert folyamatosan foglalt a gép.

## 1||∑Cj

Munkaigény szerint növekvő sorrendben adogatjuk a feladatokat a gépnek. p1≤p2≤...≤pn. Az algoritmust *SPT*-nek (Shortest Processing Time) hívjuk.

Az SPT ütemezés optimális, mert véges sok sorrend létezik és egy nem SPT ütemezés javítható. Ha az ütemezés J1, ..., Jn sorrendben veszi a feladatokat, ahol valamelyik i-re pi>pi+1, akkor a két feladatot megcserélve ∑Cj nő.

## P||Cmax

A 2||Cmax feladat NP nehéz, mert visszavezethető rá a partíciós probléma. P||Cmax is NP nehéz, mert speciális esetként tartalmazza 2||Cmax-ot.

**Def.**: listás ütemezés (Graham): A joboknak vesszük egy előre rögzített sorrendjét. Ha egy gép nem dolgozik, a listában következő munkát azonnal odaadjuk neki.

**Tétel**: a listás ütemezés (2-1/m)-approximálja a Pm||Cmax feladatot.

**Biz.**:

* OPT ≥ max(pi)
* OPT ≥ ∑pi/m
* Az utolsóként befejeződő k munka t-kor kezdődött, és t-ig egy gép sem állhatott ⇒ ∑i≠kpi ≥ mt
* Cmax = t+pk ≤ ∑i≠kpi/m+pk = ∑pi/m+pk(1-1/m) ≤ OPT\*(2-1/m)

**Tétel**: Az LPT (Longest Processing Time) szerinti listás ütemezés 4/3-approximációs minden m-re.

19. tétel

Graham közelítő algoritmusai a P||Cmax és a P|prec|Cmax feladatokra (biz. nélkül). A P|prec, pj = 1|Cmax feladat, Hu algoritmusa (biz. nélkül). A P2|prec, pj = 1|Cmax feladat: Coffman és Graham algoritmusa (biz. nélkül).

## P|prec|Cmax

**Tétel**: a listás ütemezés (ha van szabad gép, a sorban az első olyan jobot teszem fel rá, ami már elkezdhető) (2-1/m)-approximációs a P|prec|Cmax feladatra is.

## P|prec, pj=1|Cmax

A feladatosztály bonyolultsága

* P|prec, pj=1|Cmax NP-nehéz
* Pm|prec, pj=1|Cmax bonyolultsága ismeretlen
* P|prec, pj=1, in-tree|Cmax-re a Hu-algoritmus polinom időben optimális megoldást ad.

**Def.**: Az in-tree (be-fenyő) egy olyan irányított gyökeres fa, aminek az élei a gyökér felé vannak irányítva.

**Hu-algoritmus**:

1. A precedencia gráf minden csúcsához határozzuk meg a gyökérbe vezető út hosszát.
2. Alkalmazzuk a listás ütemezést úthossz szerint csökkenő sorrendben.

## P2|prec, pj=1|Cmax

**Coffman-Graham algoritmus**:

1. Végezzünk a precedencia gráfok tranzitív redukciót: ha ∃a→b él és ettől különböző a→b út, töröljük az élt.
2. Minden csúcshoz hozzárendelünk egy sorszámot és egy listát.
3. A nyelők az 1,2,3,... sorszámokat kapják valamilyen sorrendben, és üres lista tartozik hozzájuk.
4. Amint egy csúcs összes kimenő szomszédjának van sorszáma, kitöltjük a hozzá tartozó listát a kimenő szomszédok sorszámával csökkenő sorrendben.
5. Ha nincs kitölthető lista, a lexikografikusan legkisebb listával rendelkező számozatlan csúcs kapja a következő sorszámot.
6. A csúcs sorszámok szerint csökkenő sorrendben kell listás ütemezést végrehajtani.

**Tétel**: a Coffman-Graham algoritmus optimális ütemezést az a P2|prec, pj=1|Cmax feladatra.

20. tétel

Lokális élösszefüggőség és élösszefüggőségi szám fogalma. λ(G) meghatározása összehúzásokkal. Nagamochi és Ibaraki algoritmusa (biz. nélkül).

## Lokális élösszefüggőség

Egy gráfban pontok közötti lokális élösszefüggőség () az *u* és *v* pontok közötti éldiszjunkt utak száma. , vagyis az *u*, *v* pontokat elválasztó ponthalmaz minimális fokszáma.

## Gráf élösszefüggőségi száma

A lokális élösszefüggőségi számok közül a legkisebb, .

## Összefüggőség kiszámítása

G-ből készítünk egy hálózatot egységnyi élsúlyozással, és keressük a maximális folyamot, vagyis a minimális vágást. Két pont összehúzásával a vágás nem csökkenhet, és csak akkor nőhet, ha az összes minimális vágás a két pont között halad át.

Vegyük , ahol *Gu,v* a *G*-ből az *u* és *v* pontok összehúzásával kapott gráf. Ezt n – 1-szer kiszámolva megkapjuk *λ(G)*-t

A gráf pontjainak egy sorrendje max-vissza sorrend, ha esetén , ahol a *vi* és pontok közötti fokszám.

Ha *u* és *v* a max-vissza sorrend szerinti utolsó két pont, akkor . Ha tehát minden iterációban az összehúzandó párt a max-vissza sorrend utolsó két pontjaként választjuk, akkor a egyszerűen *v* fokszáma lesz. (Ekkor eltekinthetünk a folyam számolásától)

## Nagamochi + Ibaraki

1. készítsük el a gráf max-vissza sorrendjét. Ha , akkor legyen .
2. Ha még legalább három pontú a gráf, akkor húzzuk össze a max-vissza sorrend szerinti utolsó két pontot, GOTO 2. Egyébként STOP.

21. tétel

Minimális méretű 2-élösszefüggő, illetve 2-összefüggő részgráfok keresése: Khuller- Vishkin (éles példával) és Cheryan-Thurimella algoritmusok (biz. nélkül).

## Minimális méretű 2-élösszefüggő részgráf keresése

Adott egy *G* 2-élösszefüggő irányítatlan gráf, minden él költsége 1, és (élösszefüggési követelmény; ez a Hamilton-kör általánosítása: „dupla Hamilton-kör”). Feladat egy minimális élszámú 2-összefüggő feszítőgráf keresése *G*-ben. Látható, hogy az optimum értéke |*V*|, ha G-ben van Hamilton-kör. A feladat NP-nehéz, ezért nincs hatékony algoritmus, de közelítés van.

## Khuller-Vishkin

Mélységi bejárás (*T*), közben építjük a megoldást: *E*’. *T* éleit bevesszük *E*’-be. Továbbá ha az algoritmus *v* pontból visszalép egy teljesen bejárt részgráfból, akkor a *T*(*v*)-ből azt a kilépő élet hozzáadjuk *E*’-höz, ami nincs *T*-ben és a *T*(*v*)-n kívüli pontját először érte el a keresés.

-approximációs.

## Minimális méretű 2-összefüggő gráf

Egy 2-összefüggő gráf egy él vagy egy pont törlése esetén összefüggő marad.

## Cheryan-Thurimella

1. Minimális *F* lefogó élhalmazt keresünk (például úgy, hogy maximális párosítást keresünk, utána hozzáveszünk olyan éleket, hogy a párosításból kimaradt pontok is bekerüljenek)
2. Hagyjuk el a gráfból az *F*-hez nem tartozó éleket addig, amíg a gráf 2-összefüggő marad. (A kérdéses él akkor hagyható el, ha végpontjai között van 3 pontdiszjunkt út.) A megmaradt élek a kimenet (-approximáció mellett)

22. tétel

Gráfok 2-élösszefüggővé növelése: Plesnik algoritmusa, a minimálisan szükséges élek száma.

## Összefüggőség növelése

*G* gráfot minimális számú új él hozzáadásával 2-élösszegüggővé akarjuk tenni. *G* maximális élszámú 2-élösszefüggő részgráfjait *2-komponens*eknek nevezzük. Ha egy 2-komponensre egy elvágó él illeszkedik, akkor *levél* (ha egy sem, akkor *izolált levél*). Ha legalább 3 elvágó él illszekedik rá, akkor *belső komponens*. Két levél távolsága az őket összekötő utak által érintett belső komponensek száma

## Plesnik

Amíg a gráf nem 2-élösszefüggő, válasszunk két olyan levelet, amelyeknek a távolsága maximális. Adjunk egy új élt, ami a két levél egy-egy tetszőleges pontját köti össze.

Legyen , ahol a levelek száma, az izolált levelek száma. Nyilvánvaló, hogy minden új éllel kettővel csökken (kivéve, ha ), ezért az algoritmus élet használ, ami optimális.