



BUDAPEST MŰSZAKI ÉS GAZDASÁGTUDOMÁNYI EGYETEM
VILLAMOSMÉRNÖKI ÉS INFORMATIKAI KAR
Számítástudományi és Információelméleti Tanszék

RENDSZEROPTIMALIZÁLÁS – BMEVISZM117

Rendszeroptimalizálás

Vizsga tétesor jegyzet

Szerző:
GÁBOR Bernát

2026. március 1.

Tartalomjegyzék

| | |
|--|----------|
| Tartalomjegyzék | 3 |
| 1. Lineáris programozás | 7 |
| 1. Egerváry algoritmusa, az optimális hozzárendelés problémája | 7 |
| 1.1. Magyar módszer | 8 |
| 1.2. Egerváry algoritmusa | 9 |
| 1.3. Az algoritmus | 10 |
| 2. A lineáris programozás alapfeladata | 13 |
| 2.1. Kétváltozós feladat grafikus megoldása | 13 |
| 2.2. Fourier-Motzkin elimináció | 14 |
| 3. Farkas-lemma (két alakban), a lineáris program célfüggvény korlá- | |
| tossága | 17 |
| 3.1. 1-es alak | 17 |
| 3.2. 2-es alak | 19 |
| 3.3. Korlátosság | 20 |
| 4. A lineáris programozás dualitás tétele | 22 |
| 4.1. A lineáris programozás dualitás tétele (két alakban) | 22 |
| 4.2. LP bonyolultság | 23 |
| 5. Egészértékű programozás | 25 |
| 5.1. Egészértékű programozás | 25 |
| 5.2. A korlátozás és szétválasztás az IP feladatra | 26 |
| 5.3. Gyakorlati tanácsok | 27 |
| 6. Totálisan unimoduláris mátrixok és alkalmazásai | 29 |
| 6.1. Maximális összsúlyú párosítás IP feladatként | 30 |
| 6.2. Intervallumgráf | 30 |
| 7. A lineáris és egészértékű programozás alkalmazása hálózati folyam- | |
| problémákra | 33 |
| 7.1. Minimális költségű folyam keresése | 35 |
| 7.2. Több termékes folyam | 36 |

| | |
|---|-----------|
| 2. Matroidok | 37 |
| 8. Matroid definíciója és a rangfüggvény szubmodularitása | 37 |
| 8.1. Rangfüggvény szubmodularitása | 39 |
| 9. Mohó algoritmus matroidon és matroid megadása bázissal, duális matroidok | 41 |
| 9.1. Bázisos megadás | 42 |
| 9.2. Duális matroid | 42 |
| 9.3. A duális matroid rangfüggvénye | 43 |
| 10. Elhagyás, összehúzás, összeg, reprezentálhatóság | 45 |
| 11. Matroid osztályok | 49 |
| 11.1. Tutte tételei | 50 |
| 11.2. Seymour tétel | 50 |
| 12. Matroidok összege, k -matroid-metszet probléma és bonyolultsága | 51 |
| 12.1. Matroid metszet probléma – MMP _{k} | 52 |
| 12.2. Bonyolultság | 52 |
| 13. A k -matroid partíciós probléma – MPP _{k} | 54 |
| 13.1. Algoritmikus megoldás | 54 |
| 13.2. 2-matroid-metszet probléma | 56 |
| 14. k -polimatroid | 57 |
| 14.1. k -polimatroid matching probléma | 57 |
| 14.2. Bonyolultság | 58 |
| 14.3. Lovász–tétel | 58 |
| 3. Közelítő és ütemező algoritmusok | 59 |
| 15. NP-nehéz feladatok polinomiális esetei, additív és multiplikatív hiba | 59 |
| 15.1. Probléma osztályok | 60 |
| 15.2. Additív hiba | 62 |
| 15.3. Multiplikatív hiba | 63 |
| 16. Halmazfedés és a Steiner–fa | 65 |
| 16.1. Halmazfedés | 65 |
| 16.2. Steiner–fa probléma | 66 |
| 17. Az utazóügynök probléma | 69 |
| 17.1. Metrikus utazó ügynök | 69 |
| 18. Teljesen polinomiális approximációs séma a részösszeg problémára | 71 |
| 18.1. Részösszeg probléma | 71 |
| 19. Ütemezési algoritmusok | 75 |
| 19.1. $1\ C_{max}$ | 76 |
| 19.2. $1 prec C_{max}$ | 76 |
| 19.3. $1\ \sum C_j$ | 77 |
| 19.4. $P_2\ C_{max}$ | 77 |
| 19.5. $P\ C_{max}$ | 77 |

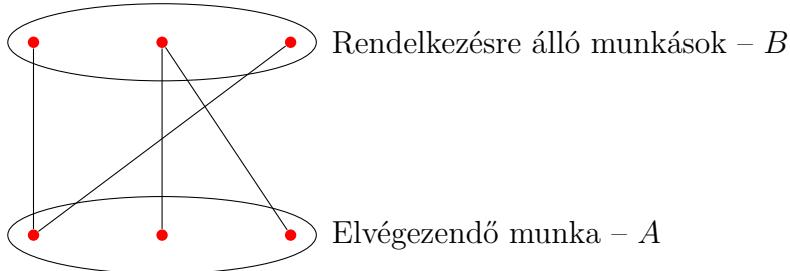
| | | |
|-------|--|----|
| 19.6. | $P prec C_{max}$ – Graham | 78 |
| 19.7. | $P prec, p_i = 1 C_{max}$ | 79 |
| 19.8. | $P in_tree, p_j = 1 C_{max}$ – Hu algoritmusá | 80 |

1. fejezet

Lineáris programozás

1. Egerváry algoritmusa, az optimális hozzárendelés problémája

Az optimális hozzárendelés problémája a következő kérdésre keresi a választ: hogyan rendelünk egymáshoz két halmaz elemeit valamilyen optimum kritérium szerint? Ilyen kritériumok lehetnek a maximális párosítás¹, minimális időigény és így tovább. Grafikusan ábrázolva, legyen:



1. ábra: Egy optimális hozzárendelés probléma

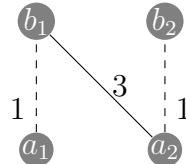
Gráf alakban megfogalmazva, adott $G = (V, E)$ gráf, ahol $V = (A, B)$ és $E = (x, y)$, ahol $x \in A$ és $y \in B$. Megkülönböztetünk 3 fajta helyzetet:

Maximális párosítás problémája Ekkor azt szeretnénk, hogy a párosítások száma minél nagyobb legyen.

Maximális összsúlyú párosítás problémája Itt létezik minden párosításnak egy jósági tényezője, egy súlya: $w : E \rightarrow \mathbb{R}$; egy adott M párosítás esetén ($M \subset E$) keressük azt, amely maximális összeget ad: $\max \{\sum_{e \in M} w(e)\}$.

¹Párosításnak nevezünk egy M élhalmazt, ha semelyik két élnek nincs közös pontja (független élhalmaz). Egy párosítás teljes, ha a gráf minden pontját lefedи.

Maximális összsúlyú teljes párosítás problémája Most olyan párosítás halmozat keressünk, amely a lehető legtöbb párosítást létrehoz, a lehető legnagyobb összeggel. A teljes párosításhoz szükséges, hogy a vizsgált ponthalmaz páros legyen. Azt, hogy ez nem ekvivalens a korábbi feladattal a 2 ábra szemlélteti. Ekkor a maximális súlyú párosítást az $a_2 - b_1$ adja, míg a maximális teljes párosítás problémájára a helyes válasz az $a_1 - b_1$ és $a_2 - b_2$ él párok.

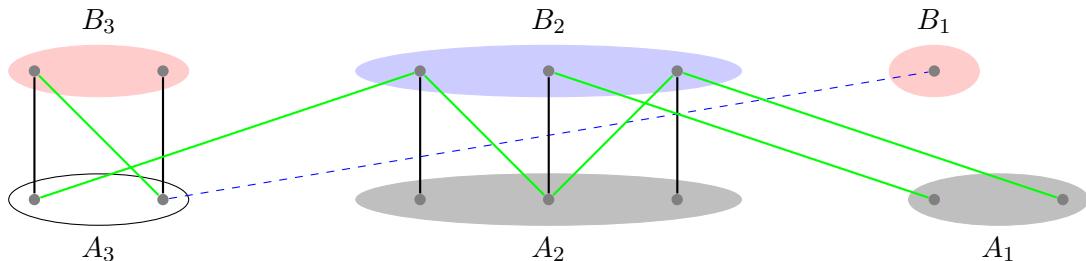


2. ábra: Maximális összsúlyú és teljes párosítás probléma

A maximális súlyú párosítás problémája visszavezethető a maximális teljes súlyú párosítás problémájára. A transzformációhoz a kevesebb csúcsú halmazt (A és B közül) egészítsük ki, hogy a két halmaz számossága megegyezzen. Majd a hiányzó és negatív súlyú éleken (a két halmazt közt) legyen $w = 0$.

1.1. Magyar módszer

Megoldás ad a maximális párosítás problémára polinomiális időben. Indulunk ki bármilyen meglévő párosításból, egy ilyet a 3 ábra mutat be.



3. ábra: Magyar módszer helyessége

Egy ilyen gráfon definiálhatóak a következők:

Alternáló út olyan él sorozat (séta) amely A -ból indul és minden második él párosításbeli.

Javító út olyan alternáló út amely B -ben végződik.

Ekkor a magyar módszer algoritmus lépései:

1. Keresünk egy javító utat.
2. Cseréljük meg az út menetén a szerepeket:
 - párosításbeli élek kivétele,
 - nem párosításbeli élek betétele.
3. Lépjünk, vissza az első lépéshoz ameddig létezik javító út.

Az algoritmus helyességének belátásához térjünk vissza a 3 ábrához, amely egy köztes állapotot szemléltet. Legyen:

(A_1) – azon csúcsok halmaza, amelyet az M párosítás nem fedett le A -ban.

(B_2) – A_1 -ből alternáló úton elérhető csúcsok halmaza,

(A_2) – B_2 -höz tartozó párosítás,

Amennyiben az algoritmus leállt, B_2 halmaz csúcsainak végpontja A_2 és A_1 halmazból induló élek, azaz B_2 lefogó ponthalmaza² az $A_1 \cup A_2$ halmaznak. Azaz elmondható, hogy A_3 és B_2 lefogó ponthalmaza a gráfnak, ugyanakkor A_3 és B_2 elem-száma megegyezik a párosítás számával. A König téTEL³ értelmében ezért a párosítás maximális ($|M| = |A_3 \cup B_2| \Rightarrow \nu(G) = \tau(G) = |\max \text{ független élhalmaz}| = |\min \text{ lefogó ponthalmaz}|$).

1.2. Egerváry algoritmusa

Az Egerváry⁴ algoritmus súlyozott páros gráfokra megadja a maximális összsúlyú teljes párosítást. Ehhez az algoritmus először definiálja a címkézés műveletet: minden csúcshoz rendel egy valós értéket ($c : V \rightarrow \mathbb{R}$) úgy, hogy minden élpár esetén ($\forall \{x, y\} \in E | x \in A, y \in B \}$ igaz a következő kifejezés: $c(x) + c(y) \geq w(e)$). Amennyiben $c(x) + c(y) = w(e)$ legyen az él „piros.” E címkézés mellett: $\sum_M w(e) \leq \sum_V c(v)$, azaz a maximális összsúlyú teljes párosítás összsúlya kisebb, mint a címkézés összege.

Bizonyítás, adódik a definícióból:

$$\sum_M w \leq \sum_M [c(x) + c(y)] = \sum_V c(v)$$

Lemma: Ha M -ben \forall él piros, a párosítás maximális.

²A lefogó ponthalmaz egy adott G (rész)gráf minden élének legalább egyik végpontját tartalmazza.

³A téTEL König Dénestől származik. Legyen egy G páros gráf, ekkor $\nu(G) = \tau(G)$ (azaz a legnagyobb független él halmaznak ugyanannyi eleme van, mint a legkisebb lefogó pont halmaznak) és ha G -ben nincs izolált pont akkor $\rho(G) = \alpha(G)$ (azaz a legkisebb lefogó él halmaz azonos méretű a legnagyobb független pont halmazzal).

⁴Egerváry Jenő

1.3. Az algoritmus

Vegyük továbbra is a 3 ábrát és az ott megfogalmazott definíciókat:

0. lépés (inicializálás):

$$c(v) = \begin{cases} \max(w) & v \in A, \\ 0 & v \in B. \end{cases}$$

1. lépés: keressük meg a maximális elem–számú párosítást a piros részgráfban javító utakkal. Legyen ez a párosítás M' . Ha ez maximális megállunk.
2. lépesben legyen:

$$\left. \begin{array}{l} x \in A_1 \cup A_2, \\ y \in B_1 \cup B_3 \\ \{x, y\} \in E \end{array} \right\} \Rightarrow \sigma = \min \{c(x) + c(y) - w(\{x, y\})\},$$

Majd:

$$c'(v) = \begin{cases} c(v) - \sigma & v \in A_1 \cup A_2, \\ c(v) + \sigma & v \in B_2, \\ c(v) & \text{másképp.} \end{cases}$$

Végül legyen $M = M'$ és $c = c'$ és folytassuk az első lépéstől.

Az algoritmus helyességének alátámasztásához be kell látnunk, hogy:

1. A nulladik lépés címkézés.
2. Létezik $\{x, y\}$ a második lépésben. Ez igaz, mert ha nem lenne akkor $A_1 \cup A_2$ bármely szomszédja B_2 -ben volna és a *Hall-feltétel*⁵ alapján nem létezne párosítás (mivel $|B_2| \geq |A_1 \cup A_2|$ nem teljesülne).
3. c' címkézés e? Ehhez figyeljük meg, hogy σ hogyan változhat:

| | $A_1 \cup A_2$ | A_3 |
|----------------|----------------|-----------|
| B_2 | 0 | $+\sigma$ |
| $B_1 \cup B_3$ | $-\sigma$ | 0 |

Ugyanakkor a $\sigma > 0$, hiszen $A_1 \cup A_2$ és $B_1 \cup B_3$ között vezető élek között nem lehet piros él. Tehát a címkézés csak egy helyen csökken (ami elronthatná a címkézést), de itt csak a maximális csökkenthető értékkel csökken, tehát a címkézés tulajdonsága megmarad.

⁵ $\forall x_0 \subseteq A$ -ra az $|N(x_0)| \geq |x_0|$ egyenlőtlenségnek teljesülnie kell, másképp nem létezik párosítás ($N x_0$ szomszédinak halmazát fedi).

4. Ahol σ minimális ott egy piros él keletkezik, ezáltal a piros részgráf is változik. Ha az $e \in B_1$ nő a párosítás mérete, mivel ha a párja A_1 -ben van akkor simán összeköthető, ha meg A_2 -ben akkor az $A_1 - B_2 - A_2 - B_1$ úton elérhető, és ez hosszabb mint az eredeti.

Ellenben, ha a piros él B_3 -beli akkor A_3 és B_2 között megszűnik egy piros él, de ez nem befolyásolja a párosítást, hiszen az új piros és B_3 -beli végpontja elérhető lesz alternáló úton, ezért átkerül az B_2 -be.

Tehát B_3 legfeljebb n lépésből elfogy. A következő lépésben B_1 -beli a piros él, tehát nő a párosítás $\Rightarrow n^2$ iteráció alatt legfeljebb megvagyunk. Egy iteráció időigénye $O(e)$ (a σ kiszámolása és $A_1 \cup A_2$ előállítása), tehát az algoritmus komplexitása $O(n^2 e)$.

2. A lineáris programozás alapfeladata, kétváltozós grafikus megoldás és Fourier-Motzkin elimináció

Egy egyenlőtlenségrendszer megoldásai közül kiválasztani azt, amely valamilyen célfüggvény szerint optimális:

$$\max(cx : Ax \leq b).$$

Az elemek méretei:

$$\begin{bmatrix} 1 & \cdots & n \end{bmatrix} \begin{bmatrix} 1 \\ \vdots \\ n \end{bmatrix} : \begin{bmatrix} 1 & \cdots & n \\ \vdots & \ddots & \vdots \\ m & \cdots & 0 \end{bmatrix} \begin{bmatrix} 1 \\ \vdots \\ n \end{bmatrix} \leq \begin{bmatrix} 1 \\ \vdots \\ m \end{bmatrix}.$$

Ha az egyenletek, vagy a feladat más alakban van megadva átalakítható az következő összefüggések alapján:

$$\begin{aligned} \alpha x \geq \beta &\Rightarrow -\alpha x \leq -\beta \\ \alpha x = \beta &\Rightarrow \begin{cases} -\alpha x \leq -\beta \\ +\alpha x \leq +\beta \end{cases} \\ \min(cx : Ax \leq b) &\Rightarrow \max((-c)x : Ax \leq b) \end{aligned}$$

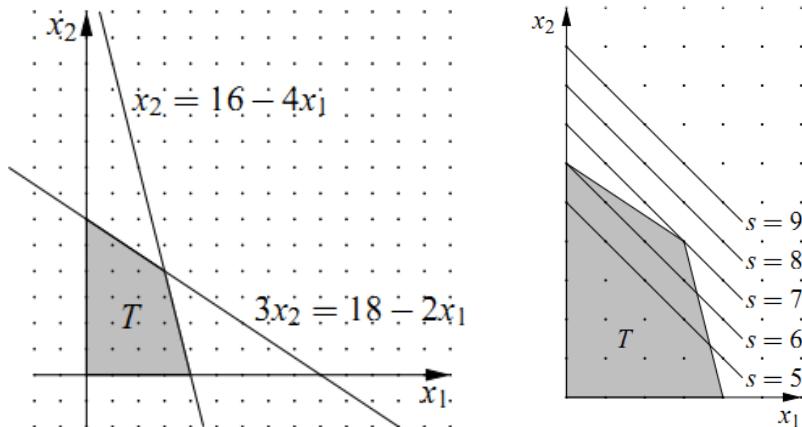
A minimumos egyenlet megoldása a maximum egyenlet megoldásának az ellen téte lesz. Ha olyan egyenletünk van, ahol szigorú egyenlőtlenség van ($<$ vagy $>$) akkor ezeket elhagyjuk, mert a valós számok halmazán nem tudunk hipersík közeli értéket keresni. A kérdés amire keressük a válaszokat:

- Létezik e az $Ax \leq b$ egyenletnek megoldása?
- cx korlátos e a megoldáshalmazon?
- Melyik x -re maximális a cx kifejezés?

2.1. Kétváltozós feladat grafikus megoldása

$\alpha x \leq \beta$ egyenlet meghatároz egy félsíket, amelyet $\alpha x = \beta$ határol. Ha a félsíkok metszete véges, egy konvex sokszöget alkotnak, amely megadja a megoldás halmazt.

Ha a célfüggvényt különböző x -re felrajzoljuk, meghatározható az optimális megoldás és az ehhez tartozó x értékek. Egy alternatív megközelítés, hogy vesszük



4. ábra: Kétváltozós feladat grafikus megoldása

a célfügvény irányvektor normálját és ennek irányából pásztázunk végig a meghatározott konvex sokszögen, a metszőpontok mentén. A megoldást az utolsó megvizsgált metszőpont adja.

2.2. Fourier-Motzkin elimináció

Ennek segítségével megvizsgálhatjuk az egyenlet megoldhatóságát: n változós egyenletet visszavezetjük $n - 1$ változóra, iteratívan, amíg egy egyváltozós egyenletrendszer nem kapunk; erre könnyű megoldhatóságot vizsgálni. A folyamat:

1. minden egyenlet $\alpha \in \mathbb{R}^+$ szorzással az alábbi alakra hozható:

$$\begin{bmatrix} 1 & A_+ \\ -1 & A_- \\ 0 & A_0 \end{bmatrix} \begin{bmatrix} x_1 & x' \end{bmatrix} \leq \begin{bmatrix} b_+ \\ b_- \\ b_0 \end{bmatrix}.$$

2. • Ha A_- üres ($= \emptyset$):

$$\begin{aligned} 1 \cdot x_1 + A_+ x' &\leq b_+ \\ 1 \cdot x_1 + \alpha x' &\leq \beta \\ x_1 &\leq \beta - \alpha x' \end{aligned}$$

Ekkor válasszuk úgy x_1 -t, olyan kicsire hogy az összes sorra e feltétel teljesüljön. Ezután A_+ teljes sorai elhagyhatóak, továbbá elég A_0 sorait vizsgálni, elhagyva a baloldalon található nulla oszlopot ($n - 1$ változós eset). Magyarul mivel nincs alsó korlát, mondhatjuk azt, hogy

választunk egy nagyon kicsi számot, ami biztos jó lesz, és innentől nem foglalkozunk az egyenlettel.

- Ha A_+ üres ($= \emptyset$):

$$\begin{aligned} -1 \cdot x_1 + \alpha x' &\leq \beta \\ x_1 &\geq \alpha x' - \beta \end{aligned}$$

Ekkor válasszuk úgy x_1 -t, olyan nagyra hogy e feltétel teljesüljön. Ezután A_- elhagyható. Az előzőhez hasonlóan olyan nagyra választjuk, hogy ne kelljen foglalkozni vele.

- Ha $A_- \neq \emptyset$ és $A_+ \neq \emptyset$ akkor legyen $i \in A_-$ egy sora, $j \in A_+$ egy sora, és képezzük az összes sorra az alábbi összeget (ez exponenciális egyenlet szaporodást jelent, e miatt az algoritmus futási ideje is exponenciális):

$$(\alpha_i + \alpha_j) \cdot x' \leq b_i + b_j$$

Ezzel visszavezettük a kérdést $n - 1$ változós estre.

3. Ha $n = 1$ és

$$\exists b_0 < 0 \Rightarrow \text{nem megoldható}$$

$$\nexists A_+ \text{ vagy } \nexists A_- < 0 \Rightarrow \text{megoldható}$$

másképp igaz, hogy $x_1 \leq \beta_+$ és $x_1 \geq \beta_- \Rightarrow$ megoldható ha $\max(-\beta_-) \leq \min(\beta_+)$

Magyarul ha az egyenletekben marad $0x_1 \leq \beta$ $\beta < 0$ formájú, akkor az egyenletek nincs megoldása. Ha $n \geq 1$ folytassuk a 2-es lépéstől.

A Gauss eliminációval szemben itt csak pozitív skalárral szorozhatjuk be a sort, az egyenlőtlenség miatt.

3. Farkas-lemma (két alakban), a lineáris program célfüggvény korlátossága

A következő egyenletekből csak a jobb vagy csak a bal egyenletrendszernek létezik egyidőben megoldása (a Farkas⁶ lemma két alakja):

1. alak:

$$\begin{array}{ll} \overbrace{Ax \leq b}^{A1} & \overbrace{yA = 0}^{A2} \\ & y \geq 0 \\ & yb < 0 \end{array}$$

2. alak:

$$\begin{array}{ll} \overbrace{Ax = b}^{B1} & \overbrace{yA \geq 0}^{B2} \\ x \geq 0 & yb < 0 \end{array}$$

Legyenek az alábbi mérettel rendelkező mátrixok:

$$\underbrace{\begin{bmatrix} y \\ 1 & \cdots & m \end{bmatrix}}_x \underbrace{\begin{bmatrix} A \\ \begin{matrix} 1 & \cdots & n \\ \vdots & \ddots & \vdots \\ m & \cdots & 0 \end{matrix} \end{bmatrix}}_A \underbrace{\begin{bmatrix} b \\ 1 \\ \vdots \\ n \\ m \end{bmatrix}}_b.$$

3.1. 1-es alak

Az első alak két egyenletrendszer nem megoldható egyidőben mert:

$$0 = 0x = (yA) \cdot x = y(Ax) \leq yb.$$

Az átcsoportosítást megtehetjük a szorzás asszociativitása miatt, míg az utolsó egyenlőtlenséget is át kell gondolni:

$$Ax \leq b \Rightarrow y(Ax) \leq yb.$$

⁶Farkas Gyula, 1902.

Szerencsére ez igaz lesz mivel Ax elemei legyenek α_i , b elemei β_i és y elemei γ_i . $\sum \gamma_i \alpha_i \leq \sum \gamma_i \beta_i \Rightarrow y(Ax) \leq y(b)$, mivel adott $y \geq 0$ feltétel (tehát $\gamma_i \geq 0$). Majd tovább futtatva a korábbi gondolatunkat $yb < 0$ amely ellentmond a kiinduló feltételeknek ($0 = yAx \leq yb < 0$).

Vizsgáljuk meg, hogy mi történik, ha A1 nem megoldható (ezt tegyük fel), következik-e ebből, hogy az A2 igen? Definiáljuk a következő C halmazt, amely egy sorvektorok halmaza, és az A2 alak egyenleteire épül:

$$C = \{z \in \mathbb{R}^{n+1} | z = y(A|b), y \geq 0\}$$

Célunk bizonyítani, hogy létezik C -ben $(0, \dots, 0, < 0)$ alakú sorvektor (ez azt jelentené, hogy az A2-nek létezik egy y megoldása), ha az A1 ($Ax \leq b$) egyenletrendszer nem megoldható. A megoldáshoz indukciót használunk, felhasználva a Fourier–Motzkin eliminációt. Lássuk be, hogy az elimináció során felhasznált műveletek (szorzás pozitív skalárral és vektorok összeadása) nem vezetnek ki a halmazból.

A halmaz zártságára az alábbi lemmák igazak ($z_1, z_2 \in C, \lambda > 0$):

- $z_1 + z_2 \in C$, mert $\begin{cases} z_1 = y_1(A|b) \\ z_2 = y_2(A|b) \end{cases} \Rightarrow z_1 + z_2 = (y_1 + y_2)(A|b)$ ahol $y_1 + y_2 \geq 0$,
- $\lambda z_1 \in C$, mert $z = y(A|b) \Rightarrow \lambda z = \lambda y(A|b)$ ahol igaz, hogy $\lambda y \geq 0$.

Bizonyítjuk, hogy az eliminációval párhuzamosan minden létre tudunk hozni ilyen elemet a C -ben, ha A1-nek nincs megoldása (mint a kezdeti halmaz elemek egy lineáris kombinációja). Kiindulunk kezdeti $(A|b)$ mátrixból, amely sorai $\in C$, mert $y_i = 1$ -el (máshol a vektor 0) felvételt nyernek C -be. Most hajtsuk végre a Fourier–Motzkin eliminációt $(A|b)$ -re, úgy, hogy a nullákat nem töröljük:

$$\begin{array}{c|c} \overbrace{\begin{array}{|c|} \hline A \\ \hline \end{array}}^{\text{sorai } \in C} & \begin{array}{c|c} b \\ \hline \end{array} \end{array} \Rightarrow \begin{array}{c|c} \overbrace{\begin{array}{|c|c|c|} \hline 0 & & \\ \hline \end{array}}^{\text{sorai } \in C} & \begin{array}{c|c} b_+ & \\ \hline b_- & \\ \hline b_0 & \\ \hline \end{array} \end{array} \Rightarrow \underbrace{\begin{array}{ccccc} 0 & 0 & 0 & +1 & b_+ \\ 0 & 0 & 0 & -1 & b_- \\ 0 & 0 & 0 & 0 & b_0 \end{array}}_{\text{sorai } \in C}$$

A zártsági lemma miatt az elimináció során a mátrix sorai C halmazon belül maradnak. Végezzük el az eliminációt újra és újra amíg egy változós alakra nem jutunk. Ha a feladat eredetileg nem volt megoldható, ezt a tulajdonságot az elimináció megtartja, a végén sem lesz az, ez meg két feltétel melett történhet meg:

- Létezik $0 \cdot x_n \leq \gamma$ egyenlet, ahol $\gamma < 0$. Ennek alakja $(0, 0, \dots, 0, < 0)$, tehát a sor $\in C$.
- $\begin{cases} +1 \cdot x_n \leq \alpha_i \\ -1 \cdot x_n \leq \alpha_j \end{cases} \Rightarrow$ ellentmondáshoz meg akkor jutunk ha $\alpha_i < -\alpha_j$.

Az i sor alakja $(0, \dots, 0, +1, \alpha_i)$, a j sor meg $(0, \dots, 0, -1, \alpha_j)$. Ennek az összege meg $(0, \dots, 0, 0, \alpha_i + \alpha_j)$ ami szintén $\in C$, mivel $\alpha_i + \alpha_j < 0$.

3.2. 2-es alak

Az, hogy a két egyenletrendszer ($B1$ és $B2$) közül csak az egyik megoldható egy időben indirekt bizonyítjuk. Legyen x, y egy-egy megoldás:

$$0 \leq \underbrace{(yA)}_{\text{sorvektor}} \quad x = y \quad \underbrace{(Ax)}_{\text{oszlopvektor}} = yb < 0$$

$\underbrace{\hspace{10em}}$
nem triviális lépés

ahol ellentmondáshoz jutunk.

Az hogy ha az egyik nem megoldható a másik igen, visszavezetéssel bizonyítjuk az 1-es alakra. Ehhez először is kibontjuk a $B1$ alakot:

$$\begin{cases} Ax = b \\ x \geq 0 \end{cases} \Rightarrow \begin{cases} Ax \leq b \\ (-Ax) \leq -b \\ (-Ex) \leq 0 \end{cases}$$

Most ennek az $A1$ alaknak felírjuk $A2$ alakját mint:

$$\left[\begin{array}{c} A \\ -A \\ -E \end{array} \right] x \leq \left[\begin{array}{c} 0 \\ 0 \\ 0 \end{array} \right] \Rightarrow \underbrace{\left[\begin{array}{c|c|c} y_1 & y_2 & y_3 \end{array} \right]}_{y'} \underbrace{\left[\begin{array}{c|c|c} A & -A & -I \end{array} \right]}_{A'} = 0, \quad \left[\begin{array}{c} y_1 \\ y_2 \\ y_3 \end{array} \right]^T \geq \left[\begin{array}{c} 0 \\ 0 \\ 0 \end{array} \right]^T, \quad \left[\begin{array}{c} b \\ -b \\ 0 \end{array} \right] < 0$$

Azt állítjuk, hogy ha ez nem megoldható, akkor a $yA \geq 0, yb \leq 0$ páros igen.

$$\begin{aligned} y_1A + y_2(-A) + y_3(-I) &= 0 \\ y_1 \cdot b - y_2 \cdot b + y_3 \cdot 0 &< 0, \\ y_1, y_2, y_3 &\geq 0 \end{aligned}$$

Tehát:

$$\left. \begin{array}{l} (y_1 - y_2) \cdot A = y_3 \\ (y_1 - y_2) \cdot b < 0 \\ y_1, y_2, y_3 \geq 0 \end{array} \right\} \xrightarrow{y=y_1-y_2} \left\{ \begin{array}{l} y \cdot A \geq 0 \\ y \cdot b < 0 \end{array} \right.$$

Ezzel felírtuk $B2$ alakot, azaz a $B1 \stackrel{?}{\Rightarrow} B2$ bizonyítását visszavezetük $A1 \Rightarrow A2$ -re.

3.3. Korlátosság

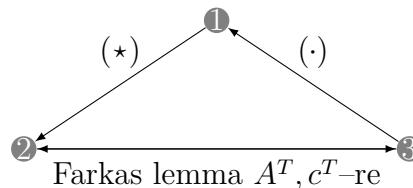
Az alábbi három kijelentés ekvivalens:

1. cx felülről korlátos az $Ax \leq b$ megoldás halmazon.

2. \nexists megoldása az $\begin{cases} Az \leq 0 \\ cz > 0 \end{cases}$ egyenletrendszernek.

3. \exists megoldása a $\begin{cases} yA = c \\ y \geq 0 \end{cases}$ egyenletrendszernek.

A tétel bizonyítását az 5 ábra szerint fogjuk belátni.



5. ábra: A korlátosság bizonyítási köre

(*) – ① \Rightarrow ②: Legyen x_0 megoldása az $Ax \leq b$ -nek és tegyük fel, hogy mégis létezik megoldása ②-nek, úgy, hogy cx felülről korlátos. Ekkor legyen $\lambda > 0$, $x_0 + \lambda z$ is megoldása az $Ax \leq b$ -nek, mert:

$$A(x_0 + \lambda z) = Ax_0 + \lambda(Az) \leq Ax_0 + 0 \leq b$$

Továbbá $c(x_0 + \lambda z) = cx_0 + \lambda(cz)$, és mivel $cz > 0$ a λ alkalmas megválasztásával ez tetszőleges naggyá tehető. Ez meg ellentmond ①-nek (mert így cx nem felülről korlátos), tehát feltevésünk hamis volt.

(\cdot) – ③ \Rightarrow ①: Legyen egy y ami teljesíti ③-at és x ami megoldása ①-nek.

$$cx = (yA)x = y(Ax) \leq yb$$

Ekkor yb a keresett felső korlát az Ax megoldás halmazán $cx - re$, tehát ③-ból következik ①.

4. A lineáris programozás dualitás tétele (két alakban), a lineáris programozás alapfeladatának bonyolultsága

4.1. A lineáris programozás dualitás tétele (két alakban)

A tétel kijelenti, hogy ha:

$$\left. \begin{array}{l} \text{LP: } \max \{cx : Ax \leq b\} \\ \text{megoldható} \\ \text{felülről korlátos} \end{array} \right\} \Rightarrow \left\{ \begin{array}{l} \text{DP: } \min \{yb : yA = c, y \geq 0\} \\ \text{(1) megoldható} \\ \text{alulról korlátos} \\ \text{(2) LP-nek } \exists \text{ maximuma} \\ \text{DP-nek } \exists \text{ minimuma} \\ \text{(3) maximum } = \text{ minimum} \end{array} \right.$$

Ezzel ekvivalens alakja az LP és a DP-nek a:

$$\begin{aligned} \text{LP: } & \max \{cx : Ax \leq b, x \geq 0\} \\ \text{DP: } & \min \{yb : yA \geq c, y \geq 0\}. \end{aligned}$$

Az (1)-est bizonyítottuk az LP korlátosságánál, a (2) és (3)-hoz felhasználjuk a következő *lemmát*:

$$\left. \begin{array}{l} Ax \leq b \text{ megoldható} \\ t \in \mathbb{R} \\ Ax \leq b \text{-nek } \nexists x \text{ megoldása, hogy } cx \geq t \end{array} \right\} \Rightarrow \{yA = c, y \geq 0\}-nak \exists \text{ olyan megoldása amelyre } yb < t.$$

A lemma *bizonyításhoz*, átfogalmazzuk a bal oldalt mint egy Farkas lemma alak: $\begin{cases} Ax \leq b \\ -cx \leq -t \end{cases}$, és tudjuk, hogy a rendszer nem megoldható (nem létezik x ami teljesíti). Alkalmazzuk rá a Farkas–lemmát, ennek második alakja felírható, mint ($\lambda \geq 0, y \geq 0$):

$$[y \ \lambda] \begin{bmatrix} A \\ -c \end{bmatrix} = yA - \lambda c = 0 \Rightarrow yA = \lambda c,$$

$$[y \ \lambda] \begin{bmatrix} b \\ -t \end{bmatrix} = yb - \lambda t < 0 \Rightarrow yb < \lambda t$$

$$\left. \begin{array}{l} yA = 0, \\ \text{Ha } \lambda = 0 \Rightarrow y \geq 0, \\ yb < 0 \end{array} \right\} \xrightarrow{\text{Farkas–lemma}} Ax \leq b \text{ nem megoldható, de ez ellentmondás}$$

az állításnak, tehát λ nem lehet nulla ($\lambda \neq 0$). Legyen $y' = \frac{1}{\lambda}y$ így az egyenletrendszer $y'A = c$, $y' \geq 0$, $y'b < t$ alakú lesz és ezzel megadtunk minden y -ra egy y' -et amely teljesíti a lemma kijelentését.

A ② bizonyításához tegyük fel, hogy \nexists maximum LP-n, ekkor legyen $t = \sup \{cx : Ax \leq b\}$ (hiszen bármely $x \subseteq R, x \neq \emptyset, x$ felülről korlátosnak \exists szuprénuma, legkisebb felső korlátja, ha még maximuma nincs is). Ha t nem maximum $\Rightarrow Ax \leq b$ -nek $\nexists cx \geq t$ teljesítő megoldása. Ekkor:

$$cx = (yA) \cdot x = y(Ax) \leq yb < t.$$

Így yb egy t -nél kisebb felső korlát cx -re, de ez ellentmondás, tehát t nem szuprénum és feltevésünk hamis volt, létezik maximum LP-n. A DP-n történő bizonyításhoz használjuk fel az LP bizonyítást, átírva a feladatott max-ra (E – egység mátrix):

$$\max \left\{ (-b)^T y^T : \begin{cases} (A^T)y^T \leq c^T \\ (-A^T)y^T \leq -c^T \\ (-E)y^T \leq 0 \end{cases} \right\}$$

A ③ bizonyításához már láttuk, hogy a

$$\max \{cx : Ax \leq b\} \leq \min \{yb : yA = c, y \geq 0\}$$

fennáll. Legyen $t = \min \{yb : yA = c, y \geq 0\}$ és tegyük fel, hogy a fenti egyenlőtlenségen nem egyenlőség áll. Tehát ekkor az LP-nek $\nexists cx \geq t$ kifejezést teljesítő megoldása, amiből következik, hogy a duális feladatnak van olyan megoldása amire $yb < t$. De ez ellentmondás a választásának, tehát feltevésünk hamis volt.

4.2. LP bonyolultság

Az LP feladat megfogalmazható, mint eldöntési probléma: $\exists?$ az $Ax \leq b$ -t kielégítő x vektor amelyre $cx \geq t$? A probléma NP-beli. Tanú egy ilyen x . Ugyanakkor coNP-beli is, a duális feladat megoldása tanú erre. x, y mérete véges.

1947 – Dantzig – Szimplex módszer Hatékony gyakorlati feladatokon, de exponenciális komplexitású.

1979 – Hacsijan – ellipszoid módszer Polinomiális komplexitás, de gyakorlatban lassú.

1984 – Karmakar – belső pontos módszer Polinomiális, hatékonyabb az ellipszoidnál, de ez is a gyakorlatban lassabb a szimplexnél.

5. Egészértékű programozás: a feladat bonyolultsága, korlátozás és szétválasztás (Branch and Bound)

5.1. Egészértékű programozás

A feladat:

$$IP : \{ \max cx : Ax \leq b, x \text{ egész} \}$$

$$DIP : \{ \min yb : yA = c, y \geq 0, y \text{ egész} \}$$

$$\max(IP) \leq \max(LP) = \min(DLP) \leq \min(DIP)$$

A feladat bonyolultságának meghatározásához fogalmazzuk át eldöntési problémára: $\exists? Ax \leq b, x \text{ egész között olyan vektor, amelyre } cx \geq t?$ Az IP feladat NP–teljes. NP–beli mert x tanú. A dualitás tétele nem alkalmazható mivel $\max(IP) < \min(DIP)$ fennállhat, tehát a probléma nem coNP–beli. *NP teljes*, mivel visszavezethető a 3–SAT problémára, és a maximális független ponthalmazra (elég az egyik is).

Maximális független ponthalmaz

$$\begin{cases} \text{input : } G(v,e) \text{ gráf, } h \text{ szám} \\ \text{output : } \alpha(G) \geq h? \\ \text{Minden csúcsra: } \forall x_i 0 \leq x_i \leq 1, x_i \in \mathbb{Z} \Rightarrow x_i = \begin{cases} 1, & \text{ha az i-edik csúcs bekerül} \\ 0, & \text{ha nem} \end{cases} \\ \text{Minden érre: } x_i + x_j \leq 1 \\ \text{max : } x_1 + x_2 + \dots + x_n \end{cases}$$

Ezzel az IP segítségével megoldottuk a max. független ponthalmazat, tehát az NP–teljes.

3–SAT

A bizonyításhoz megadjuk a visszavezetést: definiálunk egy eljárást, amely minden logikai függvényhez megad egy olyan IP problémát, melyre akkor és csak akkor igenlő a válasz, ha a függvény kielégíthető. A függvény alakja:

$$f(x_1, x_2, \dots, x_k) = \bigwedge_{i=1}^d (x_{i_1}^{e_{i_1}} \vee x_{i_2}^{e_{i_2}} \vee x_{i_3}^{e_{i_3}}).$$

Az IP probléma ismeretlenjei legyenek $z_{i=1,k}$ egyenlőtlenségek, ahol $0 \leq z_{i=1,k} \leq 1$. Továbbá minden diszjunkcióra is felírunk egy egyenlőtlenséget:

$$\begin{array}{lll}
 z_{i_1} + z_{i_2} & +z_{i_3} \geq 1 & \text{ha } x_{i_1} \vee x_{i_2} \vee x_{i_3} \\
 z_{i_1} + z_{i_2} & +(1 - z_{i_3}) \geq 1 & \text{ha } x_{i_1} \vee x_{i_2} \vee \neg x_{i_3} \\
 z_{i_1} + (1 - z_{i_2}) & +(1 - z_{i_3}) \geq 1 & \text{ha } x_{i_1} \vee \neg x_{i_2} \vee \neg x_{i_3} \\
 (1 - z_{i_1}) + (1 - z_{i_2}) & +(1 - z_{i_3}) \geq 1 & \text{ha } \neg x_{i_1} \vee \neg x_{i_2} \vee \neg x_{i_3}
 \end{array}$$

Ha f kielégíthető, akkor $z_i = 1 \Leftrightarrow x_i = \text{"igaz"}$. Ez lesz az egyenletrendszer egészértékű megoldása.

Megfordítva, ha az egyenlőtlenség-rendszert egész z_i értékkel ki tudjuk elégíteni akkor – mivel minden egész vagy 0, vagy 1 – az $x_i = \text{"igaz"} \Leftrightarrow z_i = 1$ választással f -et kielégítő értéket adtunk a logikai változónak.

Eközben a $\sum_{i=1}^k z_i$ összeg értétek valahol 0 és k között adódik, tehát c -enk $(1, 1, \dots, 1)$ választással tulajdonképpen a „Van-e a fenti egyenlőtlenségre szerek feltételeit kielégítő (egész) vektorok között olyan, melyre $cz \geq 0?$ ” kérdést tesszük fel. Az így kapott IP feladatra a válasz akkor és csak akkor igenlő, ha az f kielégíthető.

5.2. A korlátozás és szétválasztás az IP feladatra

Alakja $\max \{cx : Ax \leq b, f \leq x \leq g | x, f, g \in \mathbb{Z}^n\}$, ahol az f és g korlátok, amelyek biztosítják, hogy a feladat véges sok lépésben megoldható. A metódus kulcs-gondolata, hogy az IP feladat megoldásához először megoldjuk mint LP, ha az eredmény egész megvagyunk, egyébként tovább bontjuk két kisebb IP feladatra. Legyen $w^* = cx^*$, az egyenlőtlenség mindig $Ax \leq b$ marad.

Az algoritmus folyamata:

$$0. L = \overbrace{\left\{ \left(\underbrace{f}_{\text{alsó}}, \underbrace{g}_{\text{felső}}, \underbrace{\infty}_{w - \text{maximum értéke}} \right) \right\}}^{\text{IP részfeladatok}}, \quad \underbrace{w^* = -\infty}_{\text{eddig legjobb célfüggvény érték}}, \quad \underbrace{x^*}_{\text{legjobb megoldás}}$$

1. Ha $L = \emptyset \Rightarrow$ vége és a megoldás az aktuális w^* és x^* .

Egyébként L -ből vegyük ki IP_i feladatot $\Rightarrow (f_i, g_i, w_i)$.

2. Ha $w_i \leq w^*$, IP_i -nek a megoldása nem lehet w^* -nál jobb, folytassuk az első ponttól (ez a *Bound* lépés).

3. Oldjuk meg a relaxált IP_i feladatot (LP_i -t).

Ha \emptyset megoldás visszalépünk az első ponthoz.

Egyébként: w_i^{LP} – maximum érték, x_i – maximum hely.

4. Ha $w_i^{LP} \leq w^*$, IP_i feladat és rész feladatai maximum értéke is legfeljebb w^* , lépjünk vissza az első ponthoz.
5. Ha $w_i^{LP} > w^*$ és $x_i \in \mathbb{Z}^n$ akkor $w^* = w_i^{LP}$, $x^* = x_i$ (adat frissítés) és visszalépünk az első ponthoz.
6. Választunk egy közbenső értéket ($f_{i,j} \leq t \leq g_{i,j}$) és egy ezt meghatározó x_j elágazás változót x_i -ből (j egy pozíció x vektorban, i az IP feladat sorszáma jelöli). Ezután L -ben elhelyezzük két új feladatot: $(t+1, g, w_i^{LP})$ és (f, t, w_i^{LP}) . Lépjünk vissza az első ponthoz (*Branch* lépés).

Az algoritmus véges sok lépéssben leáll és megtalálja a feladat optimumát. A véges sok lépést f és g vektor garantálja, hiszen a kezdetben kitűzött feladatnak csak véges sok rész problémája lehet a felhasznált korlátok miatt. Az optimális részt indirekt bizonyítjuk: legyen w_0 az optimum, de az eljárás $w^* < w_0$ -t kapta.

Állítjuk, hogy L -ben mindenkorrel, hogy legyen olyan IP_i feladat amelynek az optimum értéke w_0 . Ez kezdetben fennáll, és az algoritmus egy ilyen IP feladattal a 5.-ik vagy 6.-ik lépéshez jut el. Ha az ötödikbe jut akkor az algoritmus mégis csak megtalálja a w_0 optimumot, ami ellentmondás.

Ha a hatodik lépés következik be, akkor a két keletkező IP feladat közül az egyikre teljesül, hogy a vizsgált tartományában a w_0 -hoz tartozó optimum benne van, így w_0 optimumot mindenkorrel megtaláljuk. Tehát már csak az lehet, hogy az eljárás során a lista sosem tűrült ki, de ekkor az algoritmus sose állhatott volna le, ami a keresett ellentmondás.

5.3. Gyakorlati tanácsok

- LIFO alapú új feladat választás L -ból, mert a megoldás várhatóan mélyen van a fába és ez tud legmélyebbre leggyorsabban lenyúlni. Ugyanakkor a duál szimplex módszerrel lehetséges így a korábbi számítások újra felhasználása.
- Elágazásnál a legkevésbé egész x_j -t válasszuk elágazási változónak (ez a relaxált IP feladatt megoldásának helyvektora, azaz azt az indexet ahol a vektor elemének törtrésze $\frac{1}{2}$ -hez legközelebb van), közbülső értéknek pedig ennek egész értékét (lefelé kerekítve) válasszuk. Ez azt jelenti, hogy az f és a g vektorban csak ez az indexen levő értéket modósitsuk, a többi feltétel marad a korábbi.

$$f = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}, g = \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix}, x_0 = \begin{bmatrix} 1.2 \\ 3.6 \\ 5.7 \end{bmatrix} \Rightarrow f_b = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}, g_b = \begin{bmatrix} 4 \\ 3 \\ 6 \end{bmatrix}, f_j = \begin{bmatrix} 1 \\ 4 \\ 3 \end{bmatrix}, g_j = \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix}$$

A fenti példában x_0 a talált optimum helyvektora. A 0.5-höz legközelebb a 3.6 van, ez a második változó indexet jelenti, és értéke lekerekítve 3. A többi megkötés (f és g vektor elemei) marad a korábbi. Tehát $t = 3$ és f és g vektort ennek megfelelően módosítjuk t és $t + 1$ értékekkel.

6. Totálisan unimoduláris mátrixok és alkalmazásai

Egy totálisan unimoduláris mátrixon minden négyzetes részmátrixának determinánsa 0, 1 vagy -1. Ehhez szükséges (de nem elégsges) feltétel, hogy a mátrix elemei is csak 0, 1 vagy -1 lehetnek.

$$\left. \begin{array}{l} A \text{ totálisan unimoduláris} - \text{TU} \\ b \text{ egész vektor} \\ c \text{ tetszőleges } \in \mathbb{R}^n \\ \max \{cx : Ax \leq b\} \text{ LP feladat} \\ \text{ami megoldható és felülről korlátos} \end{array} \right\} \Rightarrow \text{Az IP feladat is megoldható és az LP feladat maximuma megegyezik az IP feladat maximumával.}$$

Lemma: a totálisan unimoduláris tulajdonság megmarad, ha:

sor vagy oszlopot $\cdot(-1)$ \Rightarrow ekkor a determináns előjele változik meg.

egységvektor sor vagy oszlopként való hozzáadása \Rightarrow ha a mátrixhoz egy-ségvektort veszünk hozzá például oszlopként, és egy kiválasztott négyzetes részmátrixában ez az oszlop szerepel, akkor az új oszlop szerinti kifejtésből azonnal látszik, hogy a determináns megegyezik az eredeti mátrix egy négyzetes részmátrixának determinánsával vagy annak ellentettjével, így értéke 0, 1 vagy -1.

sor vagy oszlop ismétlés \Rightarrow ha a kiválasztott részmátrixba az eredeti is szerepel a determináns nulla lesz, ha csak az egyik marad, akkor előáll az eredeti mátrixból képzett részmátrix, amely determinánsának értéke megmarad 0, 1 vagy -1.

transzponáljuk \Rightarrow megmarad a determináns definíciójának következményeként.

Tétel. Bármely irányított gráf illeszkedés mátrixa⁷ totálisan unimoduláris..

Irányított illeszkedési mátrixnál az oszlopban 1 van, amely pontból indul az él, -1 ahova mutat; hurok-él esetében egyetlen 1-es található. A bizonyítás teljes indukcióval. Legyen $M \in \mathbb{M}_{k \times k}$ méretű részmátrix. Ha $k = 1$ az állítás nyilvánvaló.

$${}^7 B(G_{(V,E)}) = b_{i \in V, j \in E} = \begin{cases} 0 & \text{ha a } j\text{-edik él nem illeszkedik az } i\text{-edik ponthoz,} \\ +1 & \text{ha a } j\text{-edik élnek az } i\text{-edik pont kezdőpontja,} \\ -1 & \text{ha a } j\text{-edik élnek az } i\text{-edik pont a végpontja,} \\ +1 & \text{ha a } j\text{-edik él az } i\text{-edik ponthoz illeszkedő hurokél.} \end{cases}$$

Ha $k \geq 2$ és létezik egy darab oszlop, amelybe legfeljebb egy nem nulla elem van, akkor kifejtés a szerint és az indukció adja a többet (a kifejtés alapján $-1 \cdot \det M_{n-1 \times n-1}$ lesz a determináns, ami 1,0 vagy -1).

Másképp minden oszlopában egy darab +1 és egy darab -1 van, és a többi nulla. Így M sorainak összege null vektor, előállítható mint nem triviális lineáris kombináció, azaz a sorok összefüggöök, és ezért $\Rightarrow \det M = 0$.

6.1. Maximális összsúlyú párosítás IP feladatként

Tétel. Páros gráf⁸ illeszkedési mátrixa totálisan unimoduláris.

Mutasson minden él A -ból B -be. Irányított gráf illeszkedési mátrixa TU. Az eredeti illeszkedési mátrixának megkapásához B csúcsainak megfelelő sorokat -1- el szorozzuk.

Legyen x indikátor, hogy az él benne van-e párosításban, w egy tetszőleges él súlyfüggvény és B az illeszkedési mátrix (amely páros gráf lévén totálisan unimoduláris). A *feladat*, megfogalmazható, mint $\max\{wx : Bx \leq (1, \dots, 1)^T, x \geq 0\}$. Az $x \geq 0$ feltételt bevisszük B mátrixba kiegészítve azt egy $m \times m$ -es egységmátrix ellentettjével, de ez nem változtat TU tulajdonságát.

Az egyenlőtlenség így azt fejezi ki, hogy egy csúcsból legfeljebb egy kiinduló élet választunk ki, ami adja a párosítás feladatát.

$$\left. \begin{array}{l} B' = \begin{bmatrix} B \\ -I \end{bmatrix} \text{ TU} \\ b \text{ egész} \end{array} \right\} \Rightarrow x \text{ egész vektor (0 vagy 1) értékű – IP feladat}$$

A $\max\{wx : Bx \leq (1, \dots, 1)^T, x \geq 0\}$ duálisa $\min\{y(1, \dots, 1)^T : yB \geq w, y \geq 0\}$. Az y megoldás minden v csúcsra $c(v)$ címkét rendel, ahogy Egerváry Jenő párosítás algoritmusára:

$$yB \geq w \Rightarrow c(a) + c(b) \geq w(e), \forall e = \{a, b\} \in E.$$

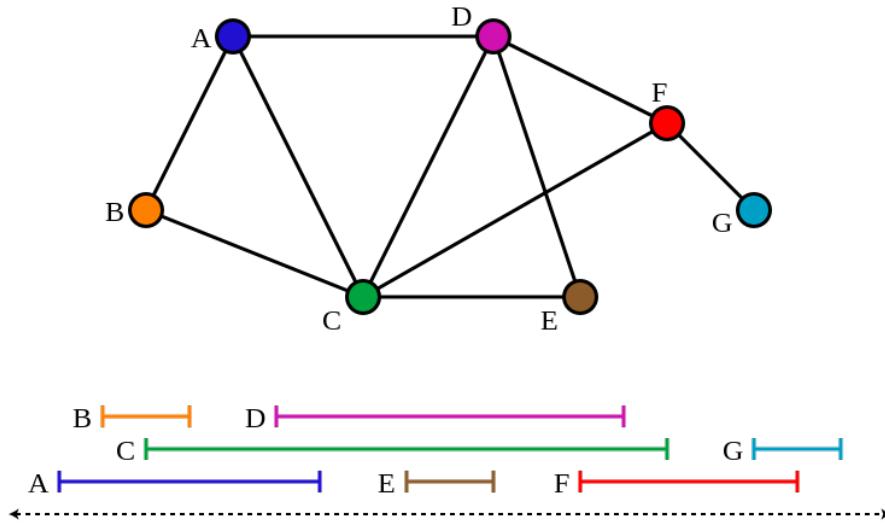
6.2. Intervallumgráf

A számegyes véges sok intervalluma alkossa egy gráf csúcshalmazát, és két csúcs akkor legyen szomszédos, ha a megfelelő intervallumok metszők; az így meghatározott gráf az intervallumgráf. Feltehető, hogy a gráfot meghatározó intervallumok

⁸Páros gráfnak nevezünk egy G gráfot, ha csúcsainak halmazát fel tudjuk úgy osztani egy A és B halmazra, hogy az összes G -beli élre teljesül, hogy az egyik végpontja A -ban van, a másik pedig B -ben.

n -re az $[1, n]$ egész végpontú, zárt részintervallumai. Legyen az $I = \{I_1, I_2, \dots, I_m\}$ intervallumrendszer.

Rendeljük ehhez $n \times m$ -es $A(I)$ mátrixot: sorai feleljenek meg az $1, 2, \dots, n$ egészeknek, oszlopai pedig az I intervallumainak. Az i -ik sor és a j -ik oszlop kereszteződésben akkor álljon 1-es, ha $i \in I_j$, és minden más helyen álljon 0.



6. ábra: Példa intervallum gráfra

Tétel. Az így definiált $A(I)$ mátrix totálisan unimoduláris.

A bizonyításhoz kiválasztunk egy tetszőleges $k \times k$ részmátrixot, majd teljes indukciót használunk az egyesek darabszáma szerint. Ha nulla darab egyesből áll a mátrix az nyilván TU. Ha van benne két oszlop, amelyben az első egyes azonos helyen áll, akkor a nagyobb egyes darabszámából kivonjuk a kisebb darabszámot. Ez az egyesek darabszámát csökkenti, de a determinánsát nem változtatja.

Ha nincs ilyen oszlop, de van csupa nulla oszlop, akkor a determináns nulla. Ha egyik sem teljesül, akkor pedig be tudjuk rendezni az oszlopokat úgy, hogy az alsóháromszög mátrixot kapunk oszlop cserével. Ez a művelet a determinánst csak előjelben változtatja meg, így a determináns 1 vagy -1 marad.

Tétel. Az intervallumgráfok tetszőleges k színre megszínezhetőek egyenletesen.

Az „egyenletesen” alatt azt értjük, hogy ha i -t tartalmazó intervallumok száma d_i , akkor ezek közül minden felhasznált szín esetén az ilyen színű intervallumok száma $\lceil \frac{d_i}{k} \rceil$ ⁹ vagy $\lfloor \frac{d_i}{k} \rfloor$ ¹⁰.

⁹felső egész rész

¹⁰alsó egész rész

Elég bebizonyítani, hogy az intervallumok közül kiválasztható néhány úgy, hogy bármely $1 \leq i \leq n$ esetén az i -t tartalmazó intervallumok között $\lceil \frac{d_i}{k} \rceil$ vagy $\lfloor \frac{d_i}{k} \rfloor$ darab kiválasztott intervallum legyen. Ekkor a kiválasztott intervallumokat megsínezük egy színnel, majd elhagyjuk őket. A megmaradt intervallumra meg ugyanaz $k - 1$ -el és így tovább.

Legyen:

$$\left. \begin{array}{l} A = A(I) \text{ intervallumokhoz rendelt mátrix} \\ d = n \text{ dimenziós vektor } i\text{-ik komponense } d_i \\ \lceil \frac{d}{k} \rceil \text{ vektor } i\text{-ik komponense } \lceil \frac{d_i}{k} \rceil \\ \lfloor \frac{d}{k} \rfloor \text{ vektor } i\text{-ik komponense } \lfloor \frac{d_i}{k} \rfloor \\ \lceil \frac{d}{k} \rceil \leq Ax \leq \lfloor \frac{d}{k} \rfloor, 0 \leq x \leq (1, \dots, 1)^T \end{array} \right\} \begin{array}{l} \text{Erre nyilvánvalóan megoldás az} \\ x = (\frac{1}{k}, \dots, \frac{1}{k})^T \text{ de tudjuk, hogy} \\ A \text{ TU} \Rightarrow \exists \text{ egészértékű megoldása} \\ \text{is.} \end{array}$$

Legyen x ez az egész értékű vektor, 0 vagy 1 elemeket tartalmaz, és az 1-es komponenseknek megfelelő oszlopok összege $\lceil \frac{d}{k} \rceil$ és $\lfloor \frac{d}{k} \rfloor$ között van. Így ezeknek az oszlopoknak megfelelő intervallumukat kiválasztva valóbban $\lceil \frac{d_i}{k} \rceil$ vagy $\lfloor \frac{d_i}{k} \rfloor$ illeszkedik i -re.

A téTEL következménye, hogy az intervallumgráfok *perfektek*¹¹ (feltéve, hogy az intervallumgráf minden részgráfja is intervallumgráf). Ez igaz, mivel legyen $k = \omega(G)$. Ekkor alkalmazva a fenti tételelt azt kapjuk, hogy a csúcsok jól színezhetőek k színnel, vagyis $\omega(G) = \chi(G)$.

¹¹Egy gráfot akkor nevezünk *perfektnek* ha rá és minden F feszített részgráfjára teljesül, hogy $\chi(F) = \omega(F)$, azaz a kromatikus szám¹² megegyezik a maximális klikkmérettel.

¹² k színnel kiszínezhető de $k - 1$ -el már nem, ahol a kiszínhezhetőség azt jelenti, hogy minden csúcsot ki lehet színezni k színnel úgy, hogy szomszédos csúcsok között nincs egyforma színű

7. A lineáris és egészértékű programozás alkalmazása hálózati folyamproblémákra

Elöször is határozzuk meg, hogy mit nevezünk folyamnak:

$$\left. \begin{array}{l} G(V, E) \text{ irányított gráf} \\ s, t \in V \text{ két kitüntet csúcs} \\ c : E \mapsto \mathbb{R}^+ \text{ nem negatív kapacitásfüggvény} \\ \rho_x(v) - v\text{-be belépő élek összege } x \text{ szerint} \\ \delta_x(v) - v\text{-ből kilépő élek összege } x \text{ szerint} \\ \text{Legyen } x : E \mapsto \mathbb{R}^+ \text{ tetszőleges függvény} \end{array} \right\} \quad \begin{array}{l} \bullet \quad x \text{ függvény akkor } folyam, \text{ ha} \\ \forall v \in V - \{s, t\} \text{-re:} \\ \rho_x(v) = \delta_x(v) \\ \bullet \quad x \text{ megengedett, ha } \forall e \in E \text{-re} \\ x(e) \leq c(e) \\ \bullet \quad \text{A folyam értéke:} \\ \delta_x(s) - \rho_x(s) = \rho_x(t) - \delta_x(t) \end{array}$$

Keressük a maximális folyamértéket ($\sum_{e \in s} x(e) = \sum_{e \in t} x(e)$), ami a forrásból kifutó és a terminálba bejövő értékek összege.

Tétel. Ford Fulkerson: A maximális folyam értéke megegyezik a minimális vágás értékével.

A fenti tételek bizonyításhoz a feladatot megfogalmazzuk LP és DP alakban, de ehhez szükségünk lesz a következő lemmára:

$$\left. \begin{array}{l} \text{Adott } x : E \mapsto \mathbb{R}^+ \\ \forall v \in V - \{s, t\} \left\{ \begin{array}{l} \delta_x(v) \leq \rho_x(v) \\ \delta_x(s) \leq \rho_x(t) \end{array} \right. \end{array} \right\} \Rightarrow x \text{ egy folyam}$$

Azaz a folyamban a pontokban érték elvesztődhet, de nem keletkezik új. Vagyunk fel $\forall v \in V - \{s, t\}$ csúcshoz egy új terminálba mutató élet, amelyekhez hozzárendeljük az $x'(e) = \rho_x(v) - \delta_x(v) \geq 0$ egyenlőtlenséget (a belépők többségbe vannak a kilépőkhöz képest, és ami eltűnik az e új élen távozik a terminálba). A többi élen maradjon meg a korábbi értékek, $x(e) = x'(e)$.

Az így konstruált x folyamhoz tartozik egy G' gráf. Mivel ez is folyam, így igaz, hogy:

$$\rho_x(t) \geq \delta_x(s) = \delta_{x'}(s) \geq \rho_{x'}(t) \geq \rho_x(t)$$

De itt csak egyenlőség állhat, ezért $\rho_{x'}(t) = \rho_x(t)$, ami csak akkor teljesülhet ha x folyam.

Az LP felíráshoz először is egészítsük ki a gráfot egy $e^* = (t, s)$ pszeudó éllel, ez lesz később majd a folyam értéke, μ .

$$\underbrace{\begin{array}{c|c} & \begin{matrix} 0 \\ \vdots \\ 0 \\ -1 \\ 1 \end{matrix} \\ \hline B & \end{array}}_{B^*} \leq \underbrace{\begin{array}{c} x \\ \mu \end{array}}_{x^*} \leq \underbrace{\begin{array}{c} 0 \\ \vdots \\ 0 \end{array}}_0$$

Az így kapott gráf illeszkedési mátrixa legyen B^* . minden $v \in V - \{s, t\}$ csúcschoz tartozik egy sor, amelyre teljesül a $b_v x \leq 0$ feltétel (azaz a belépő élek összege nagyobb, mint a kilépőké, mert a $b_v x$ ha kifejtjük $\delta_x(v) - \rho_x(v) \leq 0$ amit átrendezve kapjuk, hogy $\delta_x(v) \leq \rho_x(v)$). Továbbá a forrás és a terminál sorai alapján $\begin{cases} \delta_x(s) - \mu \leq 0 \\ \mu - \rho_x(t) \leq 0 \end{cases} \Rightarrow \delta_x(s) \leq \mu \leq \rho_x(t)$.

Ezzel felírtuk az előző lemma összes feltételét, tehát x egy folyam, ahol e e^* élen a folyam teljes átvitele átfolyik, $\delta_x(s) = \mu = \rho_x(t)$, és a folyam értéke μ .

$$\underbrace{\begin{array}{c|c} & \begin{matrix} -1 \\ 1 \\ 0 \\ 0 \end{matrix} \\ \hline B & \end{array}}_{B^*} \leq \underbrace{\begin{array}{c} x \\ \mu \end{array}}_{x^*} \leq \underbrace{\begin{array}{c} 0 \\ c \end{array}}_M$$

E szerint a maximális folyam felírható mint $\max\{(0, \dots, 0, 1)x^* : B^*x^* \leq 0; x^* \geq 0; x \leq c\}$. Az utolsó feltételt is hozzávesszük a mátrixhoz, mint az E egy egységvektor és a hozzá tartozó c rész az M vektorban.

A duális feladat megfogalmazható mint egy $\min\{yM : yB^* \geq (0, 0, \dots, 0, 1); y \geq 0\}$ optimalizálás. Láthatjuk, hogy ebből könnyedén nem tudunk egy vágás fogalmat hozzárandelni. Ezért fejezzük ki ebből y -t ($\pi(v)|w(e)$) alakban. Ekkor a duál feladat megegyezik az alábbi egyenletrendszerrel:

$$\begin{cases} (1) \pi(v) \geq 0 \text{ és } w(e) \geq 0, \\ (2) \text{ minden } e = (u, v) \text{ érre } \pi(u) - \pi(v) + w(e) \geq 0, \\ (3) \pi(t) - \pi(s) \geq 1. \end{cases}$$

A duális változói közül π a csúcsokhoz (mekkora csúcs a potenciálja), w az élekhez rendelhető (menyibe kerül nekem a szállítás az él mentén). A duális feladat célja az $m_{DLP} = \min \{\sum_{e \in E} w(e)c(e)\}$ alak minimalizálása. Állítjuk, hogy ez megegyezik a hálózati folyam minimális vágásának értékével (legyen ez m_C).

A bizonyításhoz belátjuk, hogy (A) $m_{DLP} \leq m_C$ és, hogy (B) $m_{DLP} \geq m_C$ amiből következik, hogy $m_{DLP} = m_C$.

(A) Bármely adott m_C vágáshoz könnyen készíthető olyan π és w amelyre az $m_C = \min \{\sum_{e \in E} w(e)c(e)\}$ következik. Bizonyításként adunk egy módszert ehhez: legyen S (tartalmazza a forráspontot) és T (tartalmazza a terminál csúcsot)

diszjunkt halmazok, ekkor:

$$\begin{cases} v \in S & \Rightarrow \pi(v) = 0, \\ v \in T & \Rightarrow \pi(v) = 1, \\ e \in (S, T) \text{ él} & \Rightarrow w(e) = 1, \\ \text{másképp} & \Rightarrow w(e) = 0. \end{cases}$$

Erre teljesül a $m_C = \min \{ \sum_{e \in E} w(e)c(e) \}$, amiből adódik az $m_{DLP} \leq m_C$, már csak a másik irányú egyenlőtlenséget kell beállítani.

(B) Az B^* mátrix totálisan unimoduláris, tehát y is egész értékű elemekből áll (mivel a duális feladatban, $\min\{yb : yA = c; y \geq 0\}$, szereplő c is az). Legyen adott (π, w) optimális, egészértékű megoldás, ebből kiindulva elkészítünk egy másik ugyancsak optimális megoldást, (π', w') -t, amely csak 0 vagy 1 értékű elemeket tartalmaz: $\pi'(v) = \begin{cases} 0, & \text{ha } \pi(v) \leq \pi(s), \\ 1, & \text{egyébként,} \end{cases}$ és $w'(e) = \begin{cases} 0, & \text{ha } w(e) = 0, \\ 1, & \text{ha } w(e) \geq 1. \end{cases}$

Ekkor (π', w') -re (1) és (3) teljesül. A (2)-öt indirekt bizonyítjuk. Tegyük fel, hogy egy adott $e = (u, v)$ él esetén ez nem teljesül, azaz $\pi'(u) - \pi'(v) + w'(e) < 0$. Ekkor a 0 – 1 értékűségük miatt $\pi'(u) = w'(e) = 0$ és $\pi'(v) = 1$ esetben valósulna meg. π' definíciója miatt ebben az esetben $\pi(v) > \pi(s) \geq \pi(u)$, ami ellentmondana (2)-nek, hiszen w' definíciójából adódik, hogy $w(e) = 0$ (ami azt jelentené, hogy u -ból v -be úgy nőtt a potenciál, hogy $w = 0$). A megoldás optimális mert $\sum w'(e)c(e) \leq \sum w(e)c(e)$, mivel $w'(e) \leq w(e)$. Tehát (1), (2), (3) teljesül és az így felírott π', w' egy helyes átfirás.

Most visszatérve az S és T halmazainkra legyen, $S = \{v \in V : \pi'(v) = 0\}$ és $T = \{v \in V : \pi'(v) = 1\}$. Egy adott $e = (u \in S, v \in T)$ érre $w'(e) = 1$. minden más élen w csak akkor lehet egy, ha $c(e) = 0$, mert egyébként $w'(e) = 0$ változtatása után a feltételek továbbra is fennállnának, de a $\sum w'(e)c(e)$ csökkenne. Ezért S és T között feszülő élek alkotta vágás értéke m_{DLP} , így $m_C \leq m_{DLP}$.

Ezzel egy általános bizonyítást adtunk a Ford–Fulkerson tételere, amely az így átfogalmazott alakban általánosabb kérdések megválaszolására is alkalmazható.

7.1. Minimális költségű folyam keresése

Minden élhez rendeljünk egy k költséget, amely kifejezi, hogy egy egységnyi folyam átvitele azon menyibe kerül. Ekkor kitűzhető egy olyan feladat, ami a legalább m nagyságú folyamok között keres minimális költségűt. Lineáris programozás alakban:

$$\min \{ kx : B^*x^* \leq 0, x^* \geq 0, x \leq c, \mu \geq m \}.$$

Ha az élek kapacitásai egész értékűek, akkor egész értékű folyam is választható. Ha az élek költsége is egészértékű, akkor ismert hatékony algoritmus megoldására, Ford–Fulkersontól.

7.2. Több termékes folyam

Adott egy $G = (V, E)$ irányított gráf és abban k darab pont pár: $(s_i, t_i)_{i=1,\dots,k}$. G -t továbbra is képzelhetjük út- vagy csőhálózatnak, az (s_i, t_i) pont párok pedig a k darab szállítandó termék termelő-, illetve fogyasztó helyének felelnek meg. Végül legyen adott egy $c : E \rightarrow \mathbb{R}^+$ kapacitás függvény.

A feladat egy megoldása abból áll, hogy minden élhez k darab számot fogunk hozzárendelni, megmondva, hogy az egyes termékből ott éppen mennyi halad át. Erre az esetre is alkalmazható a lineáris programozás, sőt a legalább két termékes folyamoknál már az egyetlen ismert hatékony algoritmus ($k = 1$ -re a javító utas algoritmus egész értékű megoldást ad polinomiális időben). Az egész értékű feladat viszont már itt is NP nehéz (mivel a feladatot leíró mátrix nem totálisan unimoduláris ekkor).

2. fejezet

Matroidok

8. Matroid definíciója és a rangfüggvény szubmodularitása

A matroidot definiálhatjuk a függetlenségi axiómákkal úgy, mint legyen E egy tetszőleges és véges alaphalmaz. Ugyanakkor $F \subseteq 2^E$ egy halmaz rendszer. Egy adott $M = (E, F)$ struktúra matroid, ha teljesül, hogy:

- | | |
|-----|---|
| F1) | $\emptyset \in F$ |
| F2) | $X \in F$ (F leszálló halmazrendszer – ha egy halmaz benne van X , akkor annak összes részhalmaza is benne van F -ben, ehhez szükséges az F1 axióma). |
| F3) | $\begin{cases} X, Y \in F \\ X > Y \end{cases} \Rightarrow \exists x \in X - Y, \text{ hogy } Y \cup \{x\} \in F$ (szubmodularitás, kölcsönösen bővithetőség – legyen két halmazrendszer a matroidból, ahol ez egyik számoságban több elemet tartalmaz; ha e több eleműben létezik olyan elem amely a kisebbikben nincs, akkor ezzel a kisebbiket kiegészítve szintén F -beli elemet kapunk). |

Egy $M = (E, F)$ matroidban:

független halamaznak nevezzük E alaphalmaz F -hez tartozó részhalmazait (tehát $X \subseteq E$ -re $X \in F$).

maximális független halamaznak nevezzük azt az $X \in F$ halmazt amelyhez már nem bővíthető tovább anélkül, hogy a halmaz függetlensége sérüljön.

matroid bázisai a matroid maximális független halmaza.

minimálisan összefüggő halmaznak nevezük azt az $X \in 2^E$ halmazt amely nem független, de amelyből bárhogyan veszünk el egy elemet a művelet függetlené tesszi.

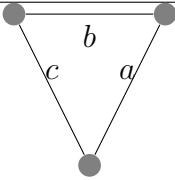
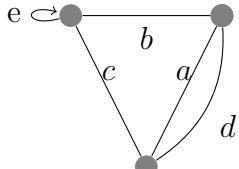
körnek nevezzük a minimálisan összefüggő halmazt.

hurok az egyelemű kör.

a halmaz rangja egy $X \subseteq E$ halmaznak az X maximális független részhalmazának mérete, elemszáma. Jelölése: $r(X)$.

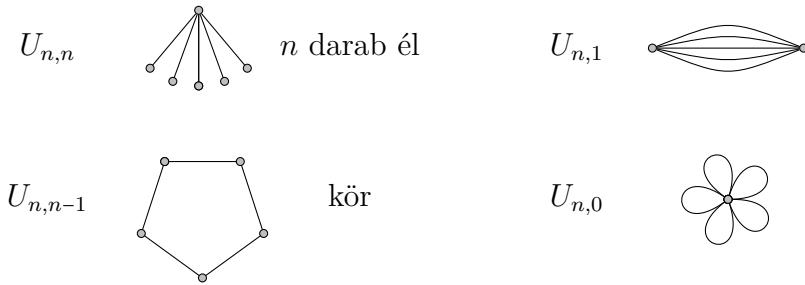
a matroid rangja az alaphalmaz rangja, azaz $r(E)$, ahol $r : 2^E \rightarrow \mathbb{Z}^+$.

Legyen $M = (E, F)$ matroid ahol $A \subseteq E$ és $X_1, X_2 \in A$ maximális független halmazok A -ban (bázisok), akkor $|X_1| = |X_2|$. A bizonyítás indirekt történik, legyen $|X_1| > |X_2| \Rightarrow \exists e \in X_1 - X_2$, ahol F3 miatt $X_1 \cup \{e\} \in F \Rightarrow X_1$ nem maximális. De ez ellentmondás, tehát a feltevésünk is hamis volt.

| Lineáris (mátrix) matroid | Grafikus matroid | Uniform matroid |
|--|---|------------------------------|
| $\begin{pmatrix} a & b \\ \underbrace{1} & \underbrace{2} \\ 0 & 0 \end{pmatrix}$ |  | $U_{2,1}$ |
| $\begin{pmatrix} a & b & c \\ \underbrace{1} & \underbrace{0} & \underbrace{1} \\ 0 & 1 & 1 \end{pmatrix}$ |  | $U_{3,2}$ |
| $\begin{pmatrix} a & b & c & d & e \\ \underbrace{1} & \underbrace{0} & \underbrace{1} & \underbrace{2} & \underbrace{0} \\ 0 & 1 & 1 & 0 & 0 \end{pmatrix}$ |  e ↪ | $ E = 5$, 32 részhalmaz |
| $\begin{pmatrix} a & b & c & d \\ \underbrace{1} & \underbrace{0} & \underbrace{1} & \underbrace{1} \\ 0 & 1 & 1 & 2 \end{pmatrix}$ | \emptyset | $U_{4,2}$ |

2.1. táblázat: Példák matroidra

Grafikus matroid egy G gráf által leírt $M = (G \text{ élei}, G \text{ -beli erdők})$. A bázis fogalma megfelel a feszítő erdőnek. Jelölése $M(G)$, körmatroidnak is nevezzük.



7. ábra: Uniform matroidok

Lineáris matroid egy mátrix által indukált matroid $M = (A$ oszlopai, A lineárisan független oszlopai). Mátrixmatroid név alatt is ismertes.

Uniform matroid egy $M = ($ tetszőleges véges halmaz, a halmaz lefeljebb k elemű részhalmazai $)$. Legyen $n = |E|$. Ekkor $U_{n,k}$ -val jelöljük az uniform matroidot. $U_{n,n}$ a teljes/szabad matroid (grafikusan egy csillag alakzat), ha összes részhalmaza független. $U_{n,0}$ a triviális matroid, amelyben csak az üres halmaz független (grafikusan minden éle egy hurok).

8.1. Rangfüggvény szubmodularitása

Bármely $X, Y \subseteq E$ -re igaz, ha $r : 2^E \mapsto \mathbb{Z}^+$ egy matroid rangfüggvénye, akkor:

$$\begin{cases} \text{R1)} & r(\emptyset) = 0 \\ \text{R2)} & r(X) \leq |X| \\ \text{R3)} & r(Y) \leq r(X), \text{ ha } Y \subseteq X \\ \text{R4)} & r(X) + r(Y) \geq r(X \cup Y) + r(X \cap Y) \end{cases}$$

Azaz (fordítva), ha r egy egészértékű függvény E részhalmazain, amelyre teljesül a fenti négy tulajdonság, akkor r egy $M = (E, F)$ matroid rangfüggvénye, ahol:

$$F = \{H : r(H) = |H|\}.$$

Az első három tulajdonság bizonyitása rögtön adódik a rang definiciójából. Az R4-re (ami a rangfüggvény szubmodularitása) legyen egy $X, Y \subseteq E$. A egy maximálisan független halmaz $X \cap Y$ -ban ($A \subseteq X \cap Y$), amely számosa α .

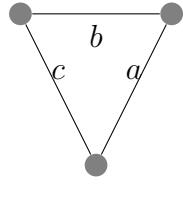
Nyilván A kiterjeszhető egy B halmazzá úgy, hogy az maximálisan független legyen $X \cup Y$ -ban. Ehhez X -ből β új elemet rakunk hozzá, és Y -ból γ darabot:

$$\begin{aligned} r(X \cap Y) &= \alpha \\ r(X \cup Y) &= \alpha + \beta + \gamma \end{aligned}$$

E számok korlátot szabnak X és Y rangjára is mint:

$$\begin{aligned} r(X) &\geq \alpha + \beta \\ r(Y) &\geq \alpha + \gamma \\ r(X) + r(Y) &\geq \underbrace{\alpha + \beta + \gamma}_{r(X \cup Y)} + \underbrace{\alpha}_{r(X \cap Y)} \\ r(X) + r(Y) &\geq r(X \cup Y) + r(X \cap Y) \end{aligned}$$

Példa arra, hogy az egyenlőség nem minden áll fent:



$$\begin{cases} X = (a, b), & r(X) = 2 \\ Y = (a, c) & r(Y) = 2 \\ X \cup Y, & r(X \cup Y) = 2 \\ X \cap Y, & r(X \cap Y) = 1 \end{cases}$$

9. Mohó algoritmus matroidon és matroid megadása bázissal, duális matroidok

$M = (E, F)$ matroid, $w : E \mapsto \mathbb{R}^+$ mekkora a maximális összsulyú független halmaz, azaz $\max\{\sum_{e \in X} w(e)\}$ ha $X \in F$, és mely ez az X ?

A mohó algoritmus tetszőleges matroid és súlyfüggvényre optimális megoldást ad a fenti kérdésre, a $\emptyset \in F$ -ből kiindulva, maximális sulyú elemeket csökenő sorrendbe véve (amik nem sérlik a halmaz függetlenségét).

A bizonyitás indirekt történik, tegyük fel, hogy az állítás nem igaz. Legyen ekkor Y az optimumot meghatározó halmaz, X pedig amit a mohó algoritmus adott. Tudjuk, hogy X és Y is maximálisan független, tehát az F3 alapján $|X| = |Y| = n$. A két halmaz elemeit rendezzük csökkenő sorrendbe, és írjuk fel őket mint $X = \{a_1, a_2, \dots, a_n\}$ és $Y = \{b_1, b_2, \dots, b_n\}$, ahol:

$$\begin{aligned} w(a_1) &\geq w(a_2) \geq \dots \geq w(a_n) \\ w(b_1) &\geq w(b_2) \geq \dots \geq w(b_n) \end{aligned}$$

A mohó algoritmus a legnagyobb sulyú elemet veszi, amely esetén a függetlenségi tulajdonság nem sérül, tehát az első elem a legnagyobb elem lesz, azaz $w(a_1) \geq w(b_1)$. Mivel a két halmaz különböző egymástól, ezért lesz egy i index amire ez a feltétel megfordul, azaz $w(b_i) > w(a_i)$, egyébbként a mohó is az optimumot kapta volna meg. Amely lépésben ez előfordul a két halmaz felírható mint:

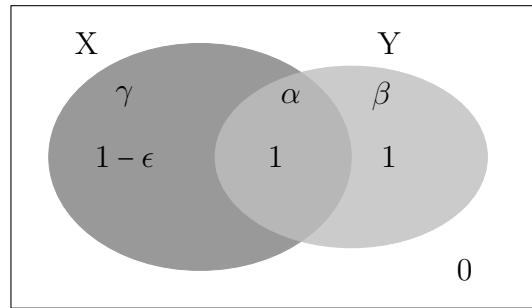
$$\begin{aligned} A &= \{a_1, \dots, a_{i-1}\}, \\ B &= \{b_1, \dots, b_i\} \end{aligned} \left. \begin{array}{l} \text{F3} \\ \hline \end{array} \right\} \Rightarrow \exists b_j \in B, \text{ amelyre } A + b_j \in F$$

Az A halmaz elemei csökkenő sorrendbe vannak rendezve, tehát $w(b_j) \geq w(b_i)$, de ugyanakkor itt $w(b_i) > w(a_i)$. De ez ellentmondás, mert ekkor a mohó algoritmus a b_j elemet választotta volna, nem pedig az a_i -t, tehát feltevésünk hamis volt, a mohó algoritmus megadja az optimális megoldást.

Ha a mohó algoritmus optimális megoldást ad és F1, F2 függetlenségi axiómák teljesülnek akkor F3 is igaz.

A bizonyitás ismét indirekt, tegyük fel, hogy a mohó algoritmus optimális megoldást ad, de F3 nem lesz igaz. Ilyenkor $X, Y \in F$, ahol $|X| > |Y|$ igaz, de $\nexists x \in X - Y$ úgy, hogy $Y \cup \{x\} \in F$. Legyen:

$$\left. \begin{array}{l} r = \frac{|Y-X|}{|X-Y|} \\ w : E \mapsto \mathbb{R}^+ \\ w(e) = \begin{cases} 1 & e \in Y, \\ r + \frac{1-r}{2} & e \in X - Y, \\ 0 & \text{másképp} \end{cases} \end{array} \right\} \Rightarrow \begin{array}{l} \text{Az algoritmus először kiválasztja } Y \text{ elemeket,} \\ \text{de ezután már csak nulla súlyút választhatna. Így az összsúly ekkor } |Y| \cdot 1. X \text{ összsúly} \\ \text{visszont } |X \cap Y| + |X - Y| \cdot (r + \frac{1-r}{2}), \text{ ami nagyobb.} \\ \text{De így egy optimálisabb megoldást kaptunk,} \\ \text{azaz a mohó nem az optimumot adta meg. De} \\ \text{ez elentmond a kiinduló feltételünknek, tehát} \\ \text{a feltvésünk, hogy F3 nem teljesül, hamis.} \end{array}$$



8. ábra: Ha a mohó algoritmus megoldása optimális, akkor F3 teljesül

9.1. Bázisos megadás

Legyen B egy matroid bázisainak halmaza, ekkor teljesül, hogy:

$$\left\{ \begin{array}{l} \text{B1)} \quad B \neq \emptyset \\ \text{B2)} \quad |X_1| = |X_2| \Leftarrow \forall X_1, X_2 \in B \\ \text{B3)} \quad \begin{array}{l} X_1, X_2 \in B \\ e_1 \in X_1 \end{array} \end{array} \right\} \Rightarrow \exists e_2 \in X_2, \text{ hogy } X_1 - e_1 + e_2 \in B$$

Fordítva, ha (E, B) egy halmazrendszer ahol a fenti három tulajdonság teljesül akkor $M = (E, F)$ matroidot alkot, ahol:

$$F = \{H : H \subseteq C \text{ valamely } C \in B\text{-re}\}$$

9.2. Duális matroid

$$\left. \begin{array}{l} M = (E, F) \text{ matroid,} \\ \text{bázisai } B = \{B_1, \dots, B_n\} \end{array} \right\} \Rightarrow \begin{array}{l} \text{A duális matroid bázisai megfogalmazható mint} \\ B^* = \{E - B_1, \dots, E - B_n\}. \text{ Ez meghatározza } F^*\text{-ot,} \\ \text{és duális matroidot mint } M^* = (E, F^*) \text{ jelöljük.} \end{array}$$

Bizonyítanunk kell, hogy az M^* matroidra teljesülnek a függetlenségi axiómák. F1 és F2 nyilvánvaló. F3–at kell belátni, azaz, hogy $X, Y \in F^*$ és $|X| > |Y|$ esetében létezik $x \in X - Y$, hogy $Y + x \in F^*$. Legyen $B_X \subseteq E - X$ és $B_Y \subseteq E - Y$ egy–egy M –beli bázis, amelyek a definíció alapján léteznek.

Ha létezik X –beli elem amely nincs benne Y –ban és B_Y –ban se, akkor ezt hozzávéve Y –hoz ismét F^* –beli halmazt kapnánk. Ha ilyen nem létezik akkor $X \subseteq B_Y \cup Y$ és

$$|B_Y \cap X| = |X - (X \cap Y)| > |Y - (X \cap Y)| \geq |B_X \cap Y|.$$

B_X és B_Y mérete megegyezik ($|B_X| = |B_Y|$), tehát $|B_Y - X| < |B_X - Y|$. Most e két halmazra alkalmazzuk F3–at, azaz $\exists z \in (B_X - Y) - (B_Y - X)$ eleme amelyre $(B_Y - X) + z$ független M –ben. Ezt a független halmazt egészítük ki bázissá, úgy, hogy B_Y –ból vesszünk hozzá új elemeket, jelöljük ezt B' –el. B' bázisába létezik elem $E - B_Y$ –ból, tehát létezik olyan B_Y –beli elem amely nincs benne B' –ben. Legyen ez az elem u . E u elemre $Y + u \subseteq E - B'$, tehát $Y + u \in F^*$. Megkaptuk az elemet amiire minden igaz lesz F3, tehát a bizonyitás teljes.

$$(M^*)^* = M$$

9.3. A duális matroid rangfüggvénye

$$r^*(X) = |X| - r(E) + r(E - X)$$

A bizonyításhoz csupán le kell vezetni:

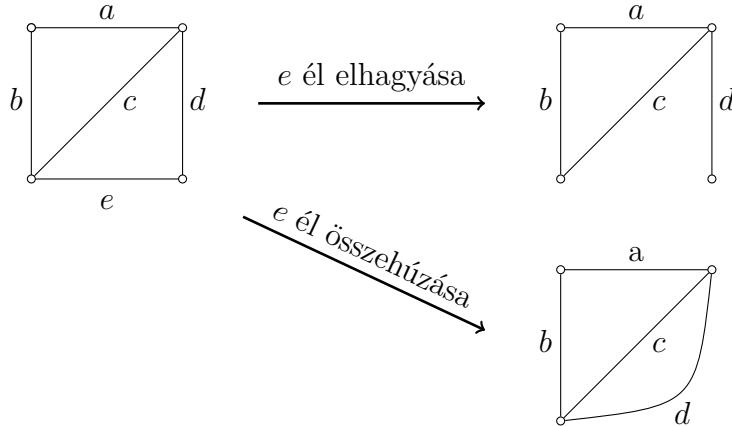
$$\begin{aligned} r^*(X) &= \max\{|X \cap Y| : Y \in F^*\} = \max\{|X \cap Y| : Y \in B^*\} \\ &= \max\{|X \cap (E - Z)| : Z \in B\} = |X| - \min\{|Z \cap X| : Z \in B\} \\ &= |X| - (r(E) - \max\{|W \cap (E - X)| : W \in B\}) \\ &= |X| - r(E) + r(E - X) \end{aligned}$$

10. Elhagyás és összehúzás. T test felett reprezentálhatóság. Matroidok direkt összege és összefüggősége.

Legyen $M = (E, F)$ matroidon $X \subseteq E$ halmazra definiáljuk a következő műveleteket:

elhagyás az $M \setminus X = (E - X, F \setminus Y)$, ahol $F \setminus Y = \{Y \subseteq E - X, Y \in F\}$ (tehát az új matroidot úgy kapjuk, hogy alaphalmazból elhagyjuk az X -beli elemeket és a halmazrendszerből meg azon elemeket amelybe ezek részt vesznek).

összehuzás az $M/X = (E - X, F/X)$, ahol az M/X rangfüggvénye felírható $r(Y) = r(X \cup Y) - r(X), Y \in E - X$ alakban (ha kivesszük az alaphalmazból az összehúzott elemeket az új alaphalmaz – Y – rangja egyenlő a teljes kezdeti alaphalmaz – $X \cup Y = E$ – és kivett alaphalmaz különbségével).



9. ábra: Elhagyás és összehúzás művelet grafikus matroidban

Az elhagyás és az összehúzás műveletek prioritása azonos, tehát felcserélhető. M matroid *minora* annak egy elhagyás és összehúzás sorozata. Bármely matroid minora előáll $N = (M \setminus A)/B$ alakban, ahol A és B diszjunkt halmazok.

$$\text{Az elhagyás és az összehúzás duális művelet: } \begin{cases} M^*/X = (M \setminus X)^*, \\ M^* \setminus X = (M/X)^*. \end{cases}$$

A bizonyításhoz elég az elsőt beláttni, mert ha az igaz, minden két oldal duálisát véve és M helyet M^* -re alkalmazva azt következik a második kijelentés. Tudjuk tehát, hogy az duális matroid összehúzásának (M^*/X) rangfüggvénye (legyen ez r_1):

$$\begin{aligned} r_1(Y) &= \underbrace{r^*(X \cup Y)}_{|X \cup Y| + r(E - X - Y) - r(E)} - \underbrace{r^*(X)}_{|X| + r(E - X) - r(E)} \\ &= |Y| + r(E - X - Y) - r(E - X) \end{aligned}$$

Ugyanakkor a matroidból X elhagyásával (M^*/X) a rangfüggvény (r_2):

$$r_2(Y) = |Y| + r(T - Y) - r(T),$$

ahol $T = E - X$ az $M \setminus X$ matroid alaphalmazza. Látjuk, hogy $r_1(Y) = r_2(Y) \Rightarrow$ minden Y -ra a két matroid megegyezik, s ezzel bizonyításunk teljes.

$M_1 = (E_1, F_1)$ matroid,
direkt összeg $M_2 = (E_2, F_2)$ matroid,
 E_1, E_2 diszjunkt, nem üres } $N = M_1 + M_2$ alaphalmaza $E_1 \cup E_2$, és
 $X \subseteq E_1 \cup E_2$ halmaz akkor független a direkt összegben, ha $X \cap E_1$ független M_1 -ben és $X \cap E_2$ független M_2 -ben.

egy matroid összefüggő ha nem áll elő matroidok direkt összegeként. A grafi-
kus matroid akkor összefüggő, ha a gráf kétszeresen összefüggő.

$M = (E, F)$ **reprezentálható** T test felett, ha bármely elem az alaphalmazból T feletti vektor.

M **matroid koordinázható** T test felett, ha létezik olyan mátrix, amelynek osz-
lopai T felett vektorok, és az ezek által meghatározott lináris matroid izomorf
 M -el.

Bármely $r = r(E), n = |E|$ matroidra létezik $A \in \mathbb{M}_{r \times n}$ mátrix amellyel leírható $M = (E, F)$ matroid (és a mátrix sorai lineárisan függetlenek). Az A mátrix meg-alkotásához r sorra van szükség, ha a matroid több elemet tartalmaz válasszunk ki r lineárisan függetlent. Ekkor a jobb oldali alakra hozható A mátrix, ettől még ugyanazt a matroidot koordinázza:

egység alaká alakít

$$\overbrace{\begin{bmatrix} 1 & \cdots & r \\ \vdots & \det \neq 0 \\ r \end{bmatrix}} \cdot \begin{bmatrix} 1 & \cdots & n \\ \vdots & A \\ r \end{bmatrix} = \begin{bmatrix} 1 & \cdots & n \\ \vdots & B \\ r \end{bmatrix} = \begin{bmatrix} 1 & \cdots & r \\ \vdots & I_r \\ r \end{bmatrix} \begin{bmatrix} 1 & \cdots & n - r \\ \vdots & A' \\ r \end{bmatrix}$$

Ha $M = (E, F)$ matroid reprezentálható T test felett akkor a duálisa (M^*) is.

A bizonyításhoz hozzuk a matroid mátrixát olyan alakra, hogy a mátrix bal oldalán egy egységmátrix alakuljon ki. Legyen ennek r sora, ekkor E_r egységmátrix M egy bázisa, míg A_0 oszlopai a többi elemek. Megpróbáljuk belátni, hogy $A' = (-A_0^T|E_{n-r})$ is reprezentálja a duális matroidot.

$$M = \begin{array}{c|c|c|c|c|c} 1 & 0 & 0 & C & & A_0 \\ \vdots & 1 & 0 & & & \\ 0 & \dots & 1 & & & \end{array} \Rightarrow M^* = \begin{array}{c|c|c|c} -C^T & 1 & 0 & 0 \\ A_0^T & 1 & 0 & 0 \\ & 1 & 0 & 0 \\ & & 1 & 0 \\ & & 0 & 1 \end{array}$$

$r-t \quad t \quad r-t \quad n-2r+t$

M matroid és a duálisának a mátrixa természetes módon megfelelhetőek egymásnak, az ábrán balról jobbra az oszlopcsoportok megfelelnek egymásnak. Válaszunk ki az M valamely bázisának megfelelő részmátrixot A -ban. Feltehető, hogy ennek első t oszlopa éppen E_r utolsó t oszlopa, míg a maradék $r-t$ oszlop A_0 első $r-t$ oszlopa. Ennek a mátrixnak determinánsa akkor nem nulla, ha C determinánsa nem nulla. Mivel B bázis, ezért C nem szinguláris.

$B^* = E - B$ elemeinek A' -ben az a mátrix felel meg, amely első $r-t$ oszlopa $-A_0^T$, a többi pedig E_{n-r} utolsó $n-2r+t$ oszlopa. Ennek a felső blokk-háromszög mátrix determinánsa akkor nem nulla, ha $-C^T$ determinánsa nem nulla. Azaz, hogy C nem szinguláris. Tehát a bázis komplementere is bázis, és fordítva, és ezért A' valóban a duálist reprezentálja.

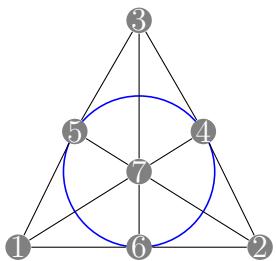
11. Matroid osztályok

A következő matroid osztályokat különböztetjük meg:

| Típus | Definició | Példa |
|------------------|--|-------------------------------|
| grafikus | ha egy G gráf által indukált (körmatroid). | K_3, K_5 |
| kografikus | a grafikus matroidok duálisa. | K_3, K_5^* |
| síkba rajzolható | ha grafikus \cap kografikus. | K_3 |
| reguláris | ha bármely test felett reprezentálható. | $K_{3,3}^*, K_5 \oplus K_5^*$ |
| bináris | ha csak a bináris test felett reprezentálható. | F_7 |
| lineáris | ha létezik test amely felett reprezentálható. | F_7^- |
| összes | ha a korábbiak közül egyik se. | $F_7 \oplus F_7^-$ |

2.2. táblázat: Matroid osztályok

Ugyanakkor igaz, az osztályokra, hogy grafikus, kografikus \subseteq reguláris \subseteq bináris \subseteq lineáris \subseteq összes.



10. ábra: A Fano matroid

A Fano matroid egy olyan matroid, amelyben az alaphalmaz mérete 7 és minden 2 elemű részhalmaz független, de minden 3 elemű csak akkor ha a 10 ábrán nincsenek egy körön vagy egy egyenesen. Ha kör megkötést elhagyjuk az így alkotott matroid az Anti-Fano matroid.

A Fano matroid pontosan azon testek felett reprezentálható amelyek karakterisztikája 2 (például a bináris matroid). Az Anti-Fano meg azon testek felett amelyek karakterisztikája nem kettő. Tehát a két matroid direkt összege semmilyen test felett nem reprezentálható, azaz nem lineáris.

Grafikus matroid bármely test felett reprezentálható (reguláris).

Indulunk ki egy n pontú gráfból, tetszőleges irányítással, rendeljünk minden ponthoz egy-egy n dimenziós egységvektort. Az élekhez rendeljük a két végpontjuk közti különbséget (az irányítás lényegtelen). Azt szeretnénk bebizonyítani, hogy egy élhalmaz akkor lineárisan független, ha az általa kifeszített részgráf körmentes.

⇒ Az élek vektorjában csak 0, 1 vagy -1 áll, tehát ezek bármely test felett reprezentálhatóak. Ha a gráfban egy kör menti éleket összeadjuk null vektort kapunk, mert a megfelelő nem nulla koordináták kiejtik egymást (az összeghez az élet hozzáadjuk ha kör körbejárási iránya megegyezik az él irányával, és kivonjuk egyébként -1 vagy -1 együtthatók).

\Leftarrow Fordítva, vegyük egy összefüggő vektorhalmazt és ennek egy olyan X nemüres részhalmazát amely olyan lineáris kombinációja ad nullát, ahol semelyik együttható nem nulla. Az élvektor azokon a koordinátákon nem nulla, amelyik pontokat összeköt.

Ahhoz, hogy a nullvektor kijöjjön minden ponthoz két darab nem nulla koordináta kell, hogy létezzen. Azaz egy pontra két él illeszkedik, amely szerint minden ilyen pont foka nagyobb mint 2. Egy adott részgráfban, ahol minden csúcs foka nagyobb mint kettő, biztosan létezik kör.

11.1. Tutte tételei

$$\begin{aligned} M \text{ matroid} & \left\{ \begin{array}{ll} \text{bináris} & \Leftrightarrow \text{nem tartalmazza minorként: } U_{4,2}. \\ \text{reguláris} & \Leftrightarrow \text{nem tartalmazza minorként: } U_{4,2}, F_7, F_7^*. \\ \text{grafikus} & \Leftrightarrow \text{nem tartalmazza minorként: } U_{4,2}, F_7, F_7^*, M^*(K_5), \\ & \quad M^*(K_{3,3}). \end{array} \right. \end{aligned}$$

11.2. Seymour téTEL

M reguláris \Leftrightarrow előáll 1 grafikus, 1 kografikus és egy R_{10} matroid példányából a direkt összeg, 2-összeg és 3-összeg műveletek segítségével.

Az R_{10} egy 5 rangú elem, egy 10 elemű halmazon: nem grafikus és nem kografikus.

$$\left(\begin{array}{ccccccccc} 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \end{array} \right)$$

11. ábra: Az R_{10}

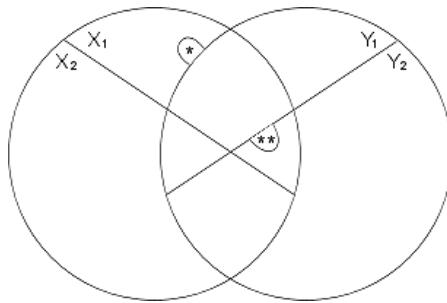
12. Matroidok összege, k-matroid-metszet probléma és bonyolultsága

$$\left. \begin{array}{l} M_1 = (E, F_1), \\ M_2 = (E, F_2) \end{array} \right\} \Rightarrow M_1 \vee M_2 = (E, F'), X \in F' \Leftrightarrow X_1, X_2 \left\{ \begin{array}{l} X = X_1 \cup X_2 \\ X_1 \in F_1 \\ X_2 \in F_2 \end{array} \right.$$

Tehát a matroid összegében azon elemek jelenek meg amelyek előállnak F_1 és F_2 -beli elemek uniójának.

Matroidok összege is matroid.

A bizonyításhoz ellenőrizzük a függetlenségi axiómákat. F_1 és F_2 nyilvánvaló, tehát csak F_3 -at látjuk be indirekt. Tegyük fel, hogy nem igaz, $X, Y \in F'$ és $|X| > |Y|$, de nem létezik $x \in X - Y$, hogy $Y + x \in F$. X és Y halmazoknak minden létezik $X = X_1 \cup X_2$ és $Y = Y_1 \cup Y_2$ ($X_1, Y_1 \in F_1, X_2, Y_2 \in F_2$) felbontása.



12. ábra: Matroid összeg bizonyitás

Válasszunk egy olyan felbontást, hogy a részhalmazok diszjunktak legyenek, $|X_1| > |Y_1|$ és az $|X_1 \cap Y_2| + |X_2 + Y_1|$ összeg minimális értéket vegyen fel. Ekkor M_1 matroidra alkalmazva az F_3 -at: létezik $x \in X_1 - Y_1$, hogy $Y_1 + x \in F_1$.

Ha $x \notin Y_2$ akkor $x + Y$ benne van az összegben ($\in F'$), ami ellentmondana az indukciós feltevésünknek. Tehát $x \in Y_2$, amiből következik, hogy létezik egy másik felbontás $Y = (Y_1 + x) \cup (Y_2 - x) = Y_1^* \cup Y_2^*$. Erre viszont a $|X_1 \cap Y_2^*| + |X_2 \cap Y_1^*|$ összeg egyel kevesebb, ami elentmond annak, hogy mi már a lehető legkisebbet választottuk korábban. Tehát a feltevésünk hamis volt, F_3 igaz.

$M = (E, F)$ matroid szabadmatroid, ha bármely részhalmaza független.

12.1. Matroid metszet probléma – MMP_k

Adott k darab matroid közös alaphalmazon: $M_i = (E, F_i)_{i=1,\dots,k}$. A kérédés amire keressük a választ, hogy létezik e $p \in \mathbb{N}$ konstansra p méretű halmaz $\cap F_i$ -ben. Az eredmény nem minden esetben lesz matroid.

$$\text{MMP}_2 \in P.$$

A bizonyitáshoz induljunk ki egy adott $M_1 = (E, F_1)$ és $M_2 = (E, F_2)$ matroid párból és legyen p a halmazrendszerök metszetébben keresett halmazok mérete. Ha $p > \min(r_1, r_2)$ akkor a válasz nem, ilyen méretű halmaz nem létezik. Másképpen csonkoljuk a két matroidot amíg rangjuk $\min(r_1, r_2, p)$ -re nem csökken:

$$\left. \begin{array}{l} M = (E, F) \text{ egy } p \text{ rangú matroid} \\ 0 \leq k \leq p \\ p \geq 1 \end{array} \right\} \begin{array}{l} F' = \{X \subseteq E : X \in F, |X| \leq k\} \\ (E, F') M \text{ csonkolat matroidja} \end{array}$$

Ezzel a problémát redukáltuk a közös bázis megkeresésére. $r_1 = r_2 = p$ -re a válasz akkor és csak akkor igenlő ha létezik ilyen méretű közös bázis, azaz ha $M_1 \vee M_2^* = (E, 2^E)$ szabad matroid.

\Rightarrow Ha M_1 és M_2 -nek van egy közös B bázisa, akkor M^* -ban $E - B$ bázis. Az $E = B \cup (E - B)$ egy felbontása az összegnek, azaz az összegben E független, tehát $M_1 \vee M_2^*$ szabadmatroid.

\Leftarrow Legyen $E = C \cup D$ egy olyan felbontása a szabadmatroidnak, hogy $C \in F_1$ és $D \in F_2^*$. Ekkor:

$$|E| \leq |C| + |D| = r_1(C) + r_2^*(D) \leq r_1(E) + r_2^*(E) = r_1(E) + |E| - r_2(E) = |E|.$$

Mivel C és D -t diszjunkt halmazoknak választottuk, C közös bázisa az M_1 és M_2 matroidnak.

12.2. Bonyolultság

MMP_k bonyolultsága ha $k \geq 3$ -ra NP teljes. A feladat NP-beli mert a p elembeli halmaz közös tanú a két halmazrendszerre.

Az NP teljességhöz figyeljük meg, hogy MMP_3 visszavezethető Hamilton út keresésére (amiről tudjuk, hogy egy NP teljes probléma). Legyen G irányított gráfban u és v között Hamilton út keresés:

$$\left. \begin{array}{l} M_1 = (E, F_1) \\ X \subseteq E \\ X \in F_1 \end{array} \right\} \Leftrightarrow X \text{ részgráfban minden pont befoka legfejlebb egy, és az } u \text{ befoka nulla.}$$

$$\left. \begin{array}{l} M_2 = (E, F_2) \\ X \subseteq E \\ X \in F_2 \end{array} \right\} \Leftrightarrow X \text{ részgráfban minden pont kifoka kisebb vagy egyenlő eggyel egy és a } v \text{ kifoka nulla.}$$

Ekkor M_3 a gráf körmatroidja, és M_1, M_2, M_3 közös $|V| - 1$ elemű bázisai G gráf irányított Hamilton útjai.

MMP_k , ha $k > 3$ NP teljes. A bizonyitáshoz az imént konstruált struktúrához vegyük hozzá $k - 3$ darab szabad matroidot.

13. A k -matroid partíciós probléma – MPP_k

Adott k darab $M_i = (E, F_i)$, ahol $i = \overline{1, k}$ matroid. A kérdés, hogy a matroidok összege ($\vee_{i=1}^k M_i$) a szabad matroidot $(E, 2^E)$ adja-e? Másnéhány megfogalmazva előáll-e az alaphalmaz E az $\cup_{i=1}^k E_i$ alakban, ha $\forall i$ -re $E_i \in F_i$. Feltehető, hogy az E_i halmazok diszjunktak, így egy partíciós problémához jutottunk.

MPP_k P -beli probléma.

$MPP_k \in \text{NP}$, mert ha a feladatnak van megoldása, az egy tanú, amely lineáris időben ellenőrizhető. $MPP_k \in \text{coNP}$, mert ha a feladatnak nem létezik megoldása, akkor adhatunk rá egy $X \subseteq E$ halmazt, ami biztosan összefüggő az összegben, azaz $\sum r_i(X) < |X|$.

13.1. Algoritmikus megoldás

Most adunk erre egy algoritmust amely megoldja a feladatott, ehhez kiindul üres halmazokból és addig bővíti azokat amíg uniójuk E nem lesz. Ha egy adott pontnál nem bővíthető tovább egyik halmaz sem, akkor az lesz a keresett ellenpélda, ami bizonyítás arra, hogy a matroidok nem partionálhatók.

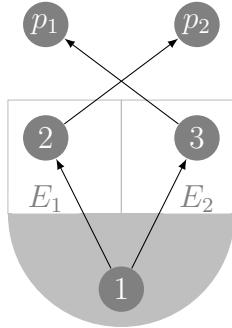


13. ábra: Példa, hogy a mohó algoritmus miért nem működik: $E_1 = \{2\}$, $E_2 = \{3\}$ állapotban nem tudunk újabb élt belevenni egyik halmazba sem. Egy jó partionálás pedig a $E_1 = \{1\}$, $E_2 = \{2, 3\}$ felosztás.

Az algoritmus egy $n + k$ pontú segédgráf dolgozik ($|E| = n$ csúcs a halmazoknak és k csúcs a partícióknak), tehát $V' = E \cup \{p_1, p_2, \dots, p_k\}$. A gráf éleit definiáljuk, mint $(\overrightarrow{xy}) \in E'$ ahol:

- $x \in E$
 - $y = \{p_i\}$
 - $x \notin E_i$
 - $E_i \cup \{x\} \in F_i$
- $$\left. \begin{array}{l} x \in E \\ y = \{p_i\} \\ x \notin E_i \\ E_i \cup \{x\} \in F_i \end{array} \right\}$$

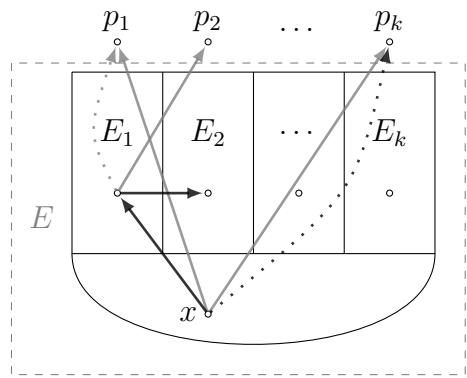
Egy ilyen él azt jelenti, hogy x -et hozzávehetjük E_i -hez a függetlenség megsérétésével ($\overrightarrow{xp_i}$ – az ábra felső részen a p_i -kbe mutató élek).



14. ábra: Segédgráf két halmazra

- $x, y \in E$
 - $x \notin E_i, y \in E_i$
 - $E_i \cup \{x\} \notin F_i,$
 - $E_i \cup \{x\} - \{y\} \in F_i$
- $\left. \begin{array}{l} \text{Egy ilyen él azt jelenti, hogy } x\text{-et hozzávehetjük } E_i\text{-hez,} \\ \text{ha } y\text{-t elhagyjuk belőle } (\overrightarrow{xy} - \text{az ábra alsó részén lévő} \\ \text{ } E_i \cup \{x\} - \{y\} \in F_i \text{ } \text{élek).} \end{array} \right\}$

0. $\forall i \ E_i = \emptyset$ kezdeti állapot ($i = \overline{1, n}$). Erre igaz, hogy $E_i \in F_i$.
1. Keressük meg a legrövidebb irányított utat $E - \cup_i E_i$ -ból $\{p_1, \dots, p_k\}$ -ba.
2. Ha létezik irányított út akkor ez meghatároz egy módosítás-sorozatot, hogy melyik E_i -t hogyan kell módosítanunk. Az út utolsó éle egy p_j -be megy, a többi E -beli elemek között, vagyis összesen 1-gyel növeljük E_i -k elem számát. Ha egy *legrövidebb* ilyen út mentén javítunk, akkor bizonyítható, hogy E_i -k a módosítások után függetlenek maradnak (mi nem bizonyítjuk).
3. Különben megállunk, nemleges a válasz, és a tanú X az $E - \cup E_i$ -ből irányított úton elérhető pontok halmaza.



15. ábra: Általános alakban

13.2. 2-matroid-metszet probléma

Adott k matroid ($M_i = (E, F_i)$, ahol $i = \overline{1, k}$) és egy p egész szám. A kérdés, hogy létezik-e F_i -nek legalább p méretű közös eleme?

Az így megfogalmazott probléma $k = 2$ -re MMP_2 komplexitása P-beli. MPP_k -ra már beláttuk, hogy P -beli, MMP_2 -őt pedig megfogalmazhatjuk mint a $M_1 \vee M_2^*$ összeg szabad matroid-e?

14. k -polimatroid

Egy $f : 2^E \mapsto \mathbb{N}$ egy k -polimatroid rangfüggvénye, ha teljesül:

$$\begin{cases} \text{KP1)} & r(\emptyset) = 0 \\ \text{KP2)} & r(\{X\}) \leq k, \text{ ekvivalens alak } r(X) \leq k|X| \\ \text{KP3)} & X \subseteq Y \Rightarrow r(X) \leq r(Y) \\ \text{KP4)} & r(X \cup Y) + r(X \cap Y) \leq r(X) + r(Y) \Leftrightarrow r(X \cup Y) \leq r(X) + k|Y| \end{cases}$$

Ha $r(\{X\}) \leq 1$, tehát $k = 1$ akkor r egy matroid rangfüggvénye.

14.1. k -polimatroid matching probléma

Ha r egy k -polimatroid rangfüggvény és $X \subseteq E$ -re $r(X) = k|X|$, akkor X -et egy k -matching halmaznak nevezzük. A k -polimatroid matching probléma felírható mint:

$$\left. \begin{array}{l} E \text{ alap halmaz} \\ r \text{ egy } k\text{-polimatroid rangfüggvény} \\ p \in \mathbb{N} \end{array} \right\} \xrightarrow{\exists X \subseteq E} \left\{ \begin{array}{l} r(X) = k|X|, \text{ tehát } X \text{ } k\text{-matching} \\ |X| \geq p \end{array} \right.$$

Figyeljük meg pár speciális esetét a problémának:

- Adott egy G gráf és egy $t \in \mathbb{N}$. $\nu(G) \geq t$? ¹ A probléma 2-polimatroid matchingként: $r(X) = |X|$ által lefedett pontok halmaza $\leq 2|X|$.
- két matroid $\left. \begin{array}{l} \\ t \in \mathbb{N} \end{array} \right\}$ létezik e $X \subseteq E$: $\left. \begin{array}{l} |X| \geq t \\ X \in F_1 \cap F_2 \end{array} \right\}$ (matroid metszet probléma)
2-polimatroid matchingént megfogalmazva: $r(X) = r_1(X) + r_2(X) \leq 2|X|$
- A két korábbi problémát összevonva $\nu(G)$ egy adott G páros gráfban?

Hogy megfogalmazzuk k -polimatroid problémának felhasználjuk a fenti két feladatot. A páros gráf két csúcshalmaza legyen A és B . M_1 grafikus matroidban két él párhuzamos ha létezik közös pontjuk A ponthalmazban. M_2 hasonlóan van definiálva B -re.

¹ $\nu(G)$ a G gráfban található független élek maximális száma.

14.2. Bonyolultság

Teljes általánosságban a probléma nem oldható meg polinom időben. Az általánoság a függvény megadási módjára vonatkozik. $\forall X \subseteq E$ -re egységnyi idő alatt $r(X)$ értéke megtudható, de ettől eltekintve a 2-polimatroid rangfüggvényéről semmit sem tudunk.

Ennek bizonyításához induljunk ki egy $|E| = 2n$ alaphalmazból és ennek egy $|E_0| = n$ részhalmazából. Továbbá legyen az alábbi két polimatroid rangfüggvény:

$$\begin{cases} \forall X \subseteq E \text{ és } |X| < n, & r_1(X) = r_2(X) = 2|X| \\ \forall X \subseteq E \text{ és } |X| > n, & r_1(X) = r_2(X) = 2n \\ \forall X \subseteq E \text{ és } |X| = n \text{ és } X = E_0, & r_1(X) = r_2(X) = 2n - 1 \\ \forall X \subseteq E \text{ és } |X| = n \text{ és } X \neq E_0, & r_1(X) = 2n - 1 \text{ és } r_2(X) = 2n \end{cases}$$

Mindkét függvény 2-polimatroid (a szubmodularitás tulajdonság bizonyítását nem részletezzük). Legyen most $p = n$. Arra a kérdésre, hogy van-e legalább p -elemű 2-matching, a válasz nyílván nemleges lesz az r_1 és igenlő az r_2 input függvény esetén.

Ha egy algoritmus teljes általánosságban meg tudja oldani a matroidpárosítási problémát, akkor a legrosszabb esetben minden $\binom{2n}{n}$ darab n elemű n elemű részhalmazra meg kell kérdeznie az r_i függvény értékét.

Hiszen, ha csak egy híján mindegyikre kérdezné meg és mindig $2n - 1$ választ kapna, akkor még nem tudhatná, hogy melyik esetünkön van szó (r_1 vagy r_2). Ez pedig már n függvényében exponenciálisan sok lépés (az analízisból ismert, hogy $\binom{2n}{n}$ aszimptotikusan egyenlő $\frac{4^n}{\sqrt{\pi n}}$ -nel).

14.3. Lovász–tételes

Vegyük egy $\mathbb{M}_{k \times 2n}$ mátrixot amely \mathbb{R} felett van koordinátázva. Oszlopai legyenek rendre $\{a_1, b_1, \dots, a_n, b_n\}$ majd definiálunk az $I = \{1, 2, \dots, n\}$ index halmazon egy r függvényt. $r(X)$ értéke egy adott $X \subseteq I$ legyen az $\cup_{i \in X} \{a_i, b_i\}$ vektorhalmaz által kifeszített altér dimenziója (hány független elem van a mátrixból kiválasztott oszlopok közül). Ilyenkor r egy 2-polimatroid rangfüggvény ($\forall i$ -re $r(\{a_i, b_i\}) = 2$ – nincs hurokél és a_i párhuzamos b_i -vel).

A matroidpárosítási probléma polinom időben megoldható, ha a 2-polimatroid rangfüggvény egy adott valós elemű M mátrixból a fent leírt módon nyerhető.

3. fejezet

Közelítő és ütemező algoritmusok

15. NP-nehéz feladatok polinomiális esetei, additív és multiplikativ hiba

Független élhalmaznak nevezzük a gráf éleinek egy olyan halmazát amelyben semelyik két élnek nincs közös pontja – $\nu(G)$ jelöli a legnagyobb független élhalmaz elemszámát.

Független ponthalmaz a csúcsok olyan részhalmaza amelyik semelyik két pont nincs közös élen – $\alpha(G)$ jelöli a legnagyobb független ponthalmaz elemszámát.

Lefogó ponthalmaz a gráf pontjainak egy halmaza amely a gráf minden élénél legalább egyik végét tartalmazza – $\tau(G)$ a legkisebb lefogó ponthalmaz elemszáma.

Lefogó élhalmaz az élek egy halmaza, ha a gráf minden pontjára legalább egy elem illeszkedik – $\rho(G)$ a legkisebb lefogó élhalmaz elemszáma.

Gallai téTEL $\forall G$ gráf ahol a gráf:

- hurokmentes $\Rightarrow \tau(G) + \alpha(G) = |V(G)|$.
- izolálpont mentes $\Rightarrow \nu(G) + \rho(G) = |V(G)|$.

König téTEL $\forall G$ gráf ahol a gráf:

- páros $\Rightarrow \tau(G) = \nu(G)$,
- izolálpont mentes $\Rightarrow \rho(G) = \alpha(G)$,

és a magyar módszer megtalálja mind a négy halmazt.

König tételel $G = (A, B, E)$ páros gráfban $\Rightarrow \chi_e(G) = \Delta(G)$, azaz az élkromatikus szám¹ megegyezik a maximális fokszámmal.

Az utóbbi bizonyítása az algoritmus megadásával történik, mely segítségével bármely G páros gráfban $\Delta(G)$ darab színnel az élek kiszínezhetőek; hatékonyan, nagy méretű gráfra is. Jelölje G éleit $E = \{e_i\}$, ahol legyen $|E| = m$. Az algoritmus egymás után színezi az éleket $\Delta(G)$ szín valamelyikével. Eközben előfordul, hogy szint kell cserélni egy élen, de azt színtelené tenni nem.

Tegyük fel, hogy e_1, \dots, e_{i-1} már ki van színezve és legyen $e_i = \{u, v\}$, ahol $u \in A$ és $v \in B$. u és v csúcsra is legfeljebb $\Delta(G)$ él illeszkedik, de ezek közül e_i még biztos, hogy színtelen. Tehát minden csúcsban van még legalább egy szabad szín (olyan színű él oda még nem kapcsolódik). Ha ez a szín azonos u és v -ben akkor használjuk azt, és készen vagyunk e_i -vel.

Ha a szín nem közös, akkor tegyük fel, hogy u szabad színe piros, míg v szabad színe kék. Legyen C a G gráf azon részgráfja amely felveszi G összes csúcsát, de az élek közül csak a piros vagy kék színűeket. C részgráfban minden minden pont foka legfeljebb 2 (hiszen minden csúcsra legfeljebb egy piros és egy kék illeszkedhet). Ekkor C diszjunkt utakból és körökből épül fel.

Ugyanakkor u és v csúcs foka C -ben csak egy lehet hiszen u -nak piros, v -nek kék a szabad színe. Ezért u és v is egy C -beli út végpontja. Legyen P_u az út melynek végpontja u . v nincs rajta ezen, mert másképp az út másik végpontja pont v lenne. Ekkor P_u páratlan sok elből állna, hiszen a gráf páros, az egyik osztályból indul (A) és másikban ér véget (B).

Tehát P_u -nak az élei váltakozó színűek, kezdete kék, tehát az utolsó él is kék színű kell, hogy legyen. Így P_u végpontja nem lehet v , mert arra nem illeszkedik kék él. Fessük át P_u kék éleit pirosra és piros éleit kékre. Ezzel a színezést nem rontjuk el, de most u -nak kék lett a szabad színe. Most már u szabad színe megegyezik $v \notin P_u$ szabad színével (kék), színezzük meg ezzel az e_i élet és megvagyunk.

15.1. Probléma osztályok

P a polinomiális időben megoldható problémák.

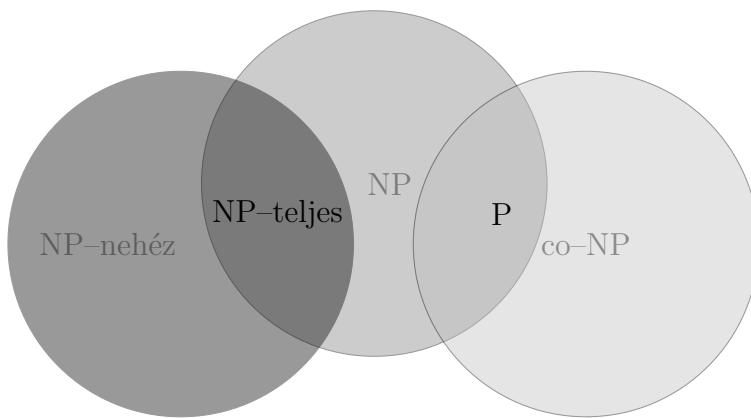
NP azon eldöntési problémák amelyre létezik az igen válaszra tanú.

co-NP azon eldöntési problémák amelyre létezik a nem válaszra tanú.

NP-nehéz \forall NP-beli probléma visszavezethető az ilyen problémára.

A 16 ábra mutatja a probléma osztályok közötti relációkat, nyílt kérdés, hogy $P = NP$ fen áll-e vagy sem.

¹A gráf élkromatikus száma k , ha a gráf élei k színnel kiszínezhető, úgy, hogy \forall 2 szomszédos él különböző színű.



16. ábra: Probléma osztályok

| Feladat | Probléma | Algoritmus |
|---|-----------|--|
| Összefüggőség | P | mélységi bejárás és szélességi bejárás |
| $\exists ?$ teljes párosítás gráfban | P | javító utak |
| $\exists ?$ 2 színezés gráfban | P | mélységi bejárás |
| $\exists ?$ 3 színezés gráfban | NP-teljes | |
| síkba rajzolhatóság | P | Kuratowski–tételes |
| klikk | NP-nehéz | |
| MAXFTL | NP-nehéz | |
| MAXFTL páros gráfban | P | magyar–módszer |
| Hamilton kör | NP-teljes | |
| utazó ügynők problémája | NP-nehéz | |
| maximális élszámú vágás keresése | NP-nehéz | |
| maximális élszámú vágás keresése, ha a hálózat síkba rajzolható | NP-nehéz | |
| $\alpha(G), \tau(G), \chi(G), \chi_e(G)$ | NP-nehéz | |
| $\alpha(G), \tau(G), \chi(G), \chi_e(G)$ ha G páros | P | |
| leghosszabb irányított út | NP-nehéz | |
| leghosszabb irányított út páros gráfban | P | mélységi bejárás |

15.2. Additív hiba

Legyen egy $f(X)$ minimalizálandó célfüggvény X_I halmazon, amely I bemeneten értelmezett. Egy algoritmus c additív hibával közelítve old meg egy minimalizálási problémát, ha bármely I -re polinom időben ad egy $y_I \in X_I$ megoldást, amire teljesül, hogy:

$$f(y_I) \leq \underbrace{(\min_{X \in X_I} f(X))}_{\text{optimum}} + \underbrace{c}_{\text{hiba}}.$$

Például határozzuk meg egy G gráfban (amelybe nincs hurokél, többszörös él és irányított él) az *élkromatikus számot*. Ekkor tudjuk, hogy $\Delta(G) = k$ és a Vizing téTEL² alapján az élkromatikus szám k vagy $k+1$. Hogy melyik azt eldöntení egy NP-teljes feladat, de a téTEL bizonyítása alapján polinom időben megkapható egy $k+1$ színezés, azaz 1 additív hibájú az algoritmus.

Egy másik példa a *kromatikus szám* (színezés) keresése egy gráfban. Az öt szín téTEL alapján polinomiális időben beszínezhető bármely gráf. Tehát egy adott síkgráfban az optimális színezés legyen 1 ha az üres, 2 ha az páros, és 5 másképpen. E utolsó esetben $\chi(G) \geq 3$, tehát a hiba ≤ 2 .

De például egy gráfban a *leghosszabb út megkeresésének* problémája (amely NP-teljes, hiszen a Hamilton-út keresését magába foglalja) semmilyen additív c konstanssal közelítő algoritmussal nem oldható meg (ha csak P=NP nem teljesül).

Ennek belátásához tegyük fel, hogy lenne egy olyan A_c algoritmus, amely tetszőleges G gráfban megadna egy olyan kört, amelynek hossza legfeljebb c -vel kisebb a leghosszabb út hosszánál. Osszuk fel G minden élét c darab új ponttal, és a kapott gráfot jelöljük H -val. Beláthatjuk, hogy:

$$H\text{-ban leghosszab kör hossza} = (c+1) \cdot G\text{-ben leghosszab kör hossza}$$

Tehát, H -ban leghosszab kör hossza egy $c+1$ -el osztható egész szám. Most futassuk le A_c algoritmust H -ra, és a kapott eredményt kerekitsük fel a legközelebbi $c+1$ -el osztható számhoz. Ezzel megadtuk H -beli maximális kör hosszát, és innen G -beli is kifejezhető. Ez egy hatékony algoritmus lenne pontos leghosszabb út megkeresésére. De ennek létezése elentmond annak, hogy a feladatt NP-nehéz, tehát indirekt bizonyítottuk állításunk.

²A Vizing-tétel alsó és felső korlátot ad egy egyszerű gráf élkromatikus számára. A téTEL Vagyim Georgijevics Vizing 1964-ben bizonyította be. A téTEL szerint egy egyszerű gráf élkromatikus száma legfeljebb egyel nagyobb a maximális fokszámánál, azaz ha a gráf minden csúcsában k -nál kevesebb él találkozik, akkor ki tudjuk színezni az éleit legfeljebb k színnel: $\Delta(G) \leq \chi_e(G) \leq \Delta(G) + 1$.

15.3. Multiplikatív hiba

Egy algoritmus $k > 1$ multiplikatív hibával közelítve old meg egy minimalizálási problémát, ha $\forall I$ bemenetre polinom időben ad egy $y_I \in X_I$ megoldást amire:

$$f(y_I) \leq \underbrace{k}_{\text{hiba}} \cdot \underbrace{\left(\min_{X \in X_I} f(X) \right)}_{\text{optimum}}.$$

Például keressük egy G gráfban a *minimális lefogó ponthalmazt* ($\tau(G)$). Válasszuk ki a független élek maximális rendszerét (ν), minden két végpontját tekintve egy 2ν elemű ponthalmazt kapunk. Definíció szerint legfeljebb annyi független él választható ki, mint a lefogó pontok minimális száma ($\nu \leq \tau$), tehát egy $k = 2$ multiplikatív hibával rendelkező algoritmusról van szó.

A *maximális páros részgráf* megállapításához keressük az n pontú gráf olyan szétosztását, ahol a két pontosztály között a legtöbb él halad. Ez a feladat egy NP–nehéz probléma. Indulunk ki egy tetszőleges kettéosztásból és, ha egy pont áthelyezése növeli a vágás élszámát akkor azt helyezzük át.

E naiv algoritmus a maximális vágás legalább felét kiadja, tehát ez is egy $k = 2$ multiplikatív hibával rendelkező algoritmus. Az algoritmus lépésszám becslése $\max_k \{k \cdot (n - k)\} \approx \frac{n^2}{4}$, ami polinom idejű, tehát teljesíti a követelményeket.

16. Halmazfedés és a Steiner–fa

16.1. Halmazfedés

A minimális lefogó ponthalmaz visszavezethető halmazfedésre, legyen a gráf élei a lefedendő alaphalmaz és a lefedő halmazokat pedig a csúcsok (így egy lefedő halmaz elemei a csúcsból kiinduló élek – alaphalmaz elemek). minden lefedő halmaz költsége legyen egységnnyi. Tehát a halmazfedés is egy NP–nehéz feladat.

A minimális összköltségű fedés problémára adunk egy közelítő algoritmust:

$$\left. \begin{array}{l} U \text{ alaphalmaz} \\ |U| = n \\ S_1, \dots, S_k \subseteq U \\ c : \{S_i\} \mapsto \mathbb{R}^+ \text{ költségfüggvény} \end{array} \right\} \begin{array}{l} \text{Legyen } C \subseteq U \text{ az } U \text{ halmazból már lefedett elemek} \\ \text{részhalmaza. Amig } C \neq U \text{ (van fedetlen elem) vá-} \\ \text{lasszuk azt az } S_i \text{ halmazt amelyre } \frac{c(S_i)}{|S_i - C|} \text{ (az illető} \\ \text{elem hozzávétele után lefedett új elemek átlagos} \\ \text{költsége) minimális.} \end{array}$$

Legyen $p(e) = \frac{c(S_i)}{|S_i - C|}$, ahol S_i az a halmaz amely először fedte le az e elemet. Ugyanakkor legyen az elemek lefedési sorrendje e_1, e_2, \dots, e_n . Állítjuk, hogy az e_k elemek ára $p(e_k) \leq \frac{\text{OPT}}{n-k+1}$, minden $k = \overline{1, n}$ -ra.

Egy optimális fedés során bármely pillanatban igaz, hogy a még fedetlen halmazok összköltsége legfeljebb OPT értékű. Tehát ekkor létezik egy olyan elem e halmazban amely értéke legfeljebb ennek $|U - C|$ hányada. Amikor az e_k elemet lefektük $|U - C| \geq n - k + 1$.

Mivel az algoritmusban mindenkor a minimális $p(e_k)$ elemet válassza:

$$p(e_k) = \frac{c(S_i)}{|S_i - C|} \leq \frac{\text{OPT}}{|U - C|} \leq \frac{\text{OPT}}{n - k + 1}.$$

Innen meg a fedés összköltsége:

$$\sum p(e) = \frac{\text{OPT}}{1} + \frac{\text{OPT}}{2} + \dots + \frac{\text{OPT}}{n} = \text{OPT} \cdot \sum_{i=1}^n \frac{1}{i} = \text{OPT} \cdot H_n \leq \text{OPT} \cdot (\ln(n) + 1).$$

Éles példa az $U = \{u_1, \dots, u_n\}$ halmazon álljon S_i az n darab egyelemű halmazból, valamint az U -ból. A költségfüggvény:

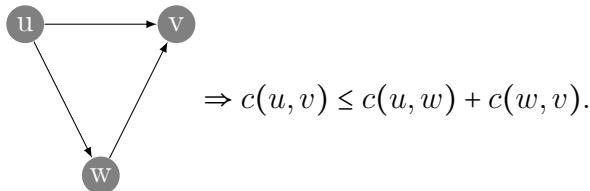
$$c : E \mapsto \mathbb{R}^+ \begin{cases} c(u_i) = \frac{1}{i}, & \forall i = \overline{1, n} \\ c(U) = 1 + \epsilon, & \epsilon \text{ kis pozitív szám} \end{cases}$$

Ekkor az algoritmus az n elemű halmazt válassza ki, pedig az optimális maga az U halmaz lenne.

16.2. Steiner-fa probléma

| | | |
|------------------------------|---|--|
| $G(V, E)$ | összefüggő gráf | Keressünk minimális költségű fát G -ben, ami az összes terminált tartalmazza és esetleg pár Steiner pontot is. |
| $c : E \mapsto \mathbb{R}^+$ | költségfüggvény | |
| $V = \{$ | T -beli terminálok, S -beli Steiner pontok | |

Ha $T = V$ (tehát a gráfban nem léteznek Steiner pontok) akkor a feladat mohó algoritmussal megoldható, egyébként meg NP-nehéz. A metrikus Steiner-fában a költségfüggvény kielégíti a háromszög egyenlőtlenséget:



Ilyenkor létezik k -approximációs algoritmus, és az általános Steiner-fa probléma visszavezethető a metrikus Steiner-fa problémára (polinom időben), úgy, hogy a k -approximáció megmarad, tehát elégseges a metrikus esetet vizsgálni.

Az átalakításhoz képezzük G' teljes gráfot V csúcs halmazon. Egy G' -beli $\{u, v\}$ él költsége – $c'(u, v)$ – legyen a legrövidebb út hossza u és v csúcsok között az eredeti gráfból (ez polinomiális időben meghatározható a Dijkstra-algoritmussal). A terminál és Steiner pont felosztás maradjon azonos az új gráfban is. minden élre G' -ben $c'(u, v) \leq c(u, v)$, tehát $\text{OPT}_{G'} \leq \text{OPT}_G$.

Legyen F' egy G' -beli Steiner-fa. F' éleit helyettesítjük G -ben a legrövidebb utakkal, hogy megkapjuk F Steiner-fát. G és G' csupán élekben különbözik, ezért F és F' is tartalmazza az összes terminált; az összköltsége a két fának megegyezik – $c(F) = c(F')$. F eszerint F' -ből polinom időben számítható, tehát helyes visszavezetés.

Ha F egy minimális költségű feszítőfa a terminálok halmazán: $c(F) \leq 2 \cdot \text{OPT}$

Vegyük egy optimális Steiner fát. Ennek éleit megduplázva egy összefüggő gráfot kapunk, ami minden terminált összeköt. Ebben keressük Euler-kört (erre létezik hatékony algoritmus). Az így kapott kör költsége 2OPT . Tegyük fel, hogy az Euler kör a $p_1, p_2, \dots, p_n = p_1$ sorrendben járja be a csúcsokat, egyes pontokat akár többször is érintve.

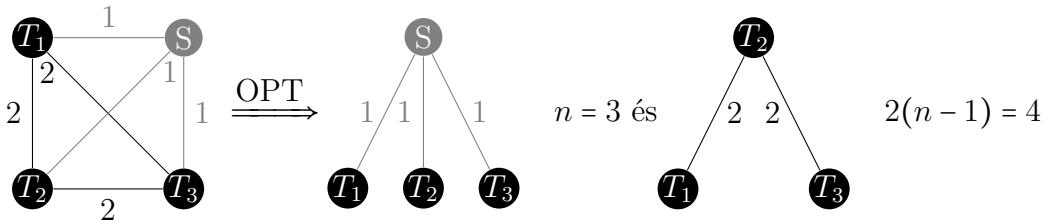
A kör mentén körbehaladva a fölösleges pontokat levágva állitsunk elő egy Hamilton kört a terminálok halmazán. Tegyük fel, hogy p_1, p_2, \dots, p_t még csupa különböző terminál, de p_{t+1} már Steiner-pont vagy egy korábbi terminál pont (a

korábbi pont indexe $i, 1 \leq i \geq t$). Legyen p_{t+j} az első olyan terminál pont amely különbözik $p_{\overline{1,t}}$ -től.

Helyettesítjük $p_t, p_{t+1}, \dots, p_{t+j}$ pont sorozatot a p_t, p_{t+j} út levágással. Ha nincs ilyen p_{t+j} akkor a p_t, p_{t+1}, \dots, p_N -ot helyettesítjük p_t, p_1 -el. Mindezt addig ismételjük amíg vissza nem jutunk p_1 terminálhoz.

A háromszög egyenlőtlenségnek köszönhetően a levágásosok nem növelik a séta költségét, ezért a kapott Hamilton kör valamennyi élét kitörölve egy Hamilton utat kapunk a terminálok halmazán, melynek költsége legfeljebb 2·OPT. Ez az út a terminálokon feszítőfa, így legalább $c(F)$ költségű. Tehát $c(F) \leq 2 \cdot \text{OPT}$. A 2 approximációs algoritmus a metrikus esetben így tehát nem más mint a terminálok halmazán minimális költségű feszítőfa keresése.

Azt, hogy az algoritmus ennél jobb approximációt biztos nem ad, az alábbi éles példa bizonyítja:



Legyen az $n + 1$ teljes gráfon n terminál és egyetlen Steiner–pont. A Steiner–pontra illeszkedő élek költsége 1, a többi él legyen 2 költségű. Itt az optimum n , a terminálokon vett minimális költségű feszítőfa költsége viszont $2(n - 1)$.

17. Az utazóügynök probléma

Adott egy G teljes gráf és egy $c : V \mapsto \mathbb{R}^+$ él súlyozás, keressünk minimális összsúlyú Hamilton–kört. A probléma NP–nehéz. Az általános utazóügynök problémára nem létezik k –approximációs algoritmus (feltéve, hogy $P \neq NP$), mivel a probléma visszavezethető egy teljes gráfban Hamilton–kör keresésére.

Legyen egy n –pontú G (nem teljes) gráf, a bemenet, hozzá tartozó G' pedig teljes gráf. Ha G -ben szomszédos két pont, akkor G -ben egységes az élsúly, egyébként kn . Ha G -ben van Hamilton–kör, akkor G' -ben a minimális összsúlyú Hamilton–kör összsúlya n , egyébként legalább $n - 1 + kn$.

Mivel ez több, mint az optimum k –szorosa, ezért nincs ilyen közelítés. Egyébként az algoritmus meg tudná különböztetni a Hamilton–körrel rendelkező és nem rendelkező gráfokat. Ugyanez kn helyett tetszőleges $f(n)$ függvényel, amely polinom időben kiszámolható, is igaz marad. Ezért a bemenettől függő approximáció sem létezik.

17.1. Metrikus utazó ügynök

A problémát egészítsük ki azzal a kikötéssel, hogy a háromszög egyenlőtlenség a pontok közti érre teljesül (csúcs \mapsto sík pontjai, élek súlya \mapsto pontok közti távolság). Erre már létezik approximációs algoritmus.

2-approximációs algoritmus

Ha egy n pontú teljes gráfban (G):

1. vesszünk egy minimális F feszítőfát³,
2. duplázzuk meg F éleit, majd keresünk egy Euler–kört,
3. az ismétlődő pontokat a körből vágjuk le, elkészítve egy Hamilton–kört.

Ha létezik a Hamilton–kör legyen ennek költsége OPT. Ebből, ha elhagyunk egy élet, kapunk egy utat amely egy Hamilton–út, értéke kisebb mint OPT és feszítőfa is. Az algoritmus első lépésében megkaptuk az optimális feszítőfát, amely e OPT-nál biztos kisebb (vagy egyenlő).

Ennek megduplázásával $2 \cdot$ OPT költségű zárt élsorozatot kaptunk, amely hosszát az elhagyások csak csökkenteni tudják (a háromszög feltétel miatt), tehát az algoritmus által talált összköltség $\leq 2 \cdot$ OPT.

³Kruskall algoritmusa – rendezzük az éleket növekvő sorrendbe, majd csökkenő sorrendbe vegyük be $n - 1$ életet amely a gráfban nem hoz létre kört.

$$\begin{aligned}
c(\text{talált Hamilton-kör}) &\leq c(\text{Euler-kör}) \\
&= 2 \cdot c(\text{minimális feszítőfa}) \\
&\leq 2 \cdot c(\text{minimális Hamilton-út}) \\
&< 2 \cdot c(\text{minimális Hamilton-kör}) \\
&= 2 \cdot OPT
\end{aligned}$$

Christofides algoritmusa

egy $\frac{3}{2}$ -approximációs algoritmus. Ez a második lépést két részre bontja:

- (a) vegyük a feszítőfa páratlan fokú pontjait. Ebből $2m$ darab van, tehát páros sok. G -ból elhagyjuk azon pontjait amelyek nincsenek benne ebben a halmazban. Legyen H az így kapott gráf.
- (b) keressünk egy minimális összsúlyú teljes párosítást H -ban. A párosítás éleit egészítsük ki (adjuk hozzá) a feszítőfához. F most egy $n - 1 + m$ élű gráf. Ebben keressük az Euler kört.

Tudjuk, hogy F súlya $\leq \text{OPT}$. Tehát ahhoz, hogy az approximáció igaz legyen a fenti procedúrával kapott párosítás legfeljebb $\frac{1}{2}\text{OPT}$ értékű lehet. Ez azt jelenti, hogy $c(H) \leq \frac{\text{OPT}}{2}$.

Ez igaz mert, ha kapunk egy minimális Hamilton-kört fessük át piros és kékre, minden második elemet. Ezzel megadtunk két teljes párosítást, ahol az olcsobb költsége $\leq \frac{\text{OPT}}{2}$. Használjuk ezt az teljes párosítást az algoritmushoz.

Míg a 2-approximáció idő igénye cn^2 , addig a Christofidesé cn^3 . Láthatjuk, hogy a jobb közelítés ára a nagyobb idő igény. Az euklideszi utazó ügynők problémára nem létezik teljes approximációs séma, de a metrikus utazó ügynők problémára igen.

A gyakorlatban gyakran heurisztikus megközelítéseket használunk, amelyekre nem létezik bizonyítás, hogy matematikailag helyes, de gyakorlatban jó eredményt ad. Ilyen a Lin–Kernighan módszer, amelyel 1 – 2% –os megközelítés is elérhető.

18. Teljesen polinomiális approximációs séma a részösszeg problémára

Egy probléma polinomiális approximációs sémával közelíthető, ha $\forall \epsilon > 0$ -ra van rá $(1 + \epsilon)$ -approximáció. Ez nem mindig elég, mert ha a közelítő algoritmus lépésszáma például $2^{\frac{1}{\epsilon}} n^4$, még mindig exponenciálisan hosszú ideig fut. Ilyenkor ugyanis rögzített ϵ -ra polinomiális az algoritmus, tehát a feltételnek megfelel.

Egy probléma *teljesen* polinomiális approximációs sémával közelíthető, ha $\forall \epsilon > 0$ -ra van rá $(1 + \epsilon)$ -approximáció, ami $\frac{1}{\epsilon}$ -ban is polinomiális. Például a metrikus utazó ügynök problémára nincs P approximációs séma, de az euklideszi utazó ügynök problémára van teljesen P approximációs séma.

18.1. Részösszeg probléma

$$A = \{a_1, a_2, \dots, a_n\} \\ a_{i=1, \dots, n}, t \in F \quad \left. \right\} \Rightarrow \exists? B \subseteq A, \text{ hogy } \sum b_i = t$$

Partíciós problémáról beszélünk, ha $t = \frac{\sum a_i}{2}$. Mindkét esetben a feladat NP-teljes. A részösszeg optimizálási probléma, ha azon B -re vagyunk kíváncsiak amelyre $\sum b_i$ maximális és $\sum b_i \leq t$. A feladat NP nehéz, mert a részösszeg probléma visszavezethető rá: ha találunk olyan B halmazt, amire $\sum b_i = t$, akkor igen a válasz a részösszeg problémára, különben nem. A feladat nem NP-beli, mert nem eldöntési probléma, tehát nem is NP-teljes.

A részösszeg probléma probléma teljesen polinomiális sémával közelíthető.

A bizonyítás egy konkrét algoritmus lesz:

1. Pontos megoldást adó algoritmus

Ez az algoritmus gyakorlatilag nem más, mint egy „brute-force” megoldó, amely minden lehetséges részösszeget kiszámol. Legyen $a_1 \leq a_2 \leq \dots \leq a_n$. Definiáljunk két halmazsorozatot:

$$L'_i = \{I + a_i | I \in L_i\}, \text{ ahol } L_0 = \{0\} \\ L_{i+1} = L_i \cup L'_i$$

Az optimális részletösszeg $\max\{I | I \in L_n \text{ és } I \leq t\}$ lesz.

Például, ha $a_1 = 3$ $a_2 = 5$ $a_3 = 7$:

$$\begin{array}{ll} L_0 = \{0\} & \\ L'_0 = \{3\} & L_1 = \{0, 3\} \\ L'_1 = \{5, 8\} & L_2 = \{0, 3, 5, 8\} \\ L'_2 = \{7, 10, 12, 15\} & L_3 = \{0, 3, 5, 7, 8, 10, 12, 15\} \end{array}$$

2. Polinomiális közelítő algoritmus

Egyrészt vegyük észre, hogy az L'_i halmazokból nincs szükségünk a t -nél nagyobb elemekre. Töröljük ki ezeket: $L'_i = (L_i + a_i) \cap [0 \dots t]$.

Ugyanakkor, ritkítuk a halmazt δ -val: legyen L a növekvő sorrendbe rendezett halmaz. Ezután a halmaz minden elemét vizsgáljuk meg, növekvő sorrendben, ha létezik kisebb elem (m) az aktuálisnál (l) amelyre $l < m(1+\delta)$ (az aktuális $1+\delta$ -szor nagyobb) akkor dobuk ki az aktuálist (l -et). A ritkítás után bármely két szomszédos elem hányadosa legalább $1+\delta$. A ritkított halmaz mérete felülről becsülhető: $|L_{\text{ritkított}}| \leq \log_{1+\delta} t + 2$.

3. Ha a fenti algoritmus során a halmazok t -nél nagyobb elemeit minden lépésben levágjuk és az eredményt $\delta = \frac{\epsilon}{2n}$ -vel ritkítjuk, $(1+\epsilon)$ -approximációt kapunk a problémára, és a lépésszám n -ben, $\log t$ -ben és $\frac{1}{\epsilon}$ -ban is polinomiális lesz.

Az $1+\epsilon$ approximációs tulajdonságot lássuk be. Ehhez elősző figyeljük meg a következő lemmát:

$$\ln(1+X) \geq \frac{X}{1+X}, \text{ ha } X \geq 0.$$

Ha $X = 0$ minden oldal nulla lesz, tehát az állítás igaz. Ha $X \geq 0$:

$$\left(\ln(1+X) - \frac{X}{1+X} \right)' = \frac{1}{1+X} - \frac{(1+X)-X}{(1+X)^2} = \frac{1}{1+X} - \frac{1}{(1+X)^2} > 0$$

Tehát a derivált pozitív, amiből következik, hogy a függvény monotonan nő, tehát a lemma igaz.

Most az algoritmus lépésszámát vizsgálva, a kapott halmaz mérete:

$$\log_{1+\delta} t = \log_{1+\frac{\epsilon}{2n}} t = \frac{\ln t}{\ln(1+\frac{\epsilon}{2n})} \leq \frac{\ln t}{\frac{\epsilon}{1+\frac{\epsilon}{2n}}} = \left(1 + \frac{\epsilon}{2n}\right) \cdot 2n \cdot \frac{\ln t}{\epsilon}$$

Az egyenlőtlenséghez felhasználtuk a lemmát $\ln(1 + \frac{\epsilon}{2n}) \geq \frac{\frac{\epsilon}{2n}}{1 + \frac{\epsilon}{2n}}$. Megfigyelhetjük, hogy a lépésszám minden n -ben, $\log t$ -ben és $\frac{1}{\epsilon}$ is polinomiális.

Az algoritmus során n darab lista összefésülését kell elvégezni, aminek a komplexitása $O(\sum |L_i|) = O(n \cdot |L_i|)$. Felhasználva a korábbi becslésünket az algoritmus minden négyzetes, $\log t$ -ben lineáris, azaz minden esetben polinomiális. Most olyan megoldást keressünk, amely az approximációs séma követelményeit teljesíti, azaz megoldása $\geq \frac{\text{OPT}}{1 + \epsilon}$.

A kapott válasz relatív hibájának kis értékére való bizonyításához vegyük figyelembe, hogy L_i lista ritkításakor legfeljebb $\frac{\epsilon}{n}$ nagyságú hibát okozunk a megmaradó képviselő érték és a ritkítás előtti érték között. i -re vonatkozó teljes indukcióval bizonyítható, hogy ezzel eredményünk az optimum $(1 + \frac{\epsilon}{2n})^n$ hányada lesz. Ez meg teljes approximáció lesz mert:

$$\underbrace{\frac{\text{OPT}}{1 + \epsilon}}_{\text{teljes approximáció}} \leq \underbrace{\frac{\text{OPT}}{(1 + \frac{\epsilon}{2n})^n}}_{\text{amennyit max rontunk}} \leq \text{kapott érték} \leq \text{OPT}.$$

Ahhoz, hogy ez igaz legyen bizonyítanunk kell, hogy $(1 + \frac{\epsilon}{2n})^n \leq 1 + \epsilon$ is igaz:

$$\begin{aligned} \left(1 + \frac{\epsilon}{2n}\right)^n &= \sum_{i=0}^n \left(\frac{\epsilon}{2n}\right)^i \cdot 1^{n-i} \cdot \binom{n}{i} = \sum_{i=0}^n \left(\frac{\epsilon}{2n}\right)^i \cdot \frac{n!}{i!(n-i)!} \leq 1 + \sum_{i=0}^n \left(\frac{\epsilon}{2n}\right)^i \cdot n^i \\ &\leq 1 + \sum_{i=1}^n \left(\frac{\epsilon}{2}\right)^i = 1 + \frac{\epsilon}{2} + \frac{\epsilon}{4} + \dots + \frac{\epsilon}{2^n} \leq 1 + \epsilon \end{aligned}$$

A bizonyítás során felhasználásra került a binomiális téTEL és a mértani sor összegképlete.

19. Ütemezési algoritmusok

Az ütemezési feladat megfogalmazható mint:

$$\left. \begin{array}{l} J_i - \text{munkák} \\ p_i - \text{megmunkálási idő} \\ w_i - \text{súly} \\ d_i - \text{határidő} \\ r_i - \text{rendelkezésre álló idő} \\ C_i - \text{befejezási iödő} \end{array} \right\} \Rightarrow + \text{ egy adott időben egy gépen egy munka folyik} \\ \text{és létezik egy célfüggvény amire nézve optimális üte-} \\ \text{mezést keresünk.}$$

Egy halmazfeladat gyors leírására a következő struktúrát használjuk:

$$\underbrace{\alpha}_{\text{gépi infó}} \quad | \quad \underbrace{\beta}_{\text{ütemezési infó}} \quad | \quad \underbrace{\gamma}_{\text{optimalizálási kritérium}}$$

gépi információ alatt a rendelkezésre álló gép számosságot értjük. Lehet:

- 1 – egy gép áll rendelkezésre.
- P_m – m darab párhuzamosan futó gép
- P – a párhuzamosan futó gépek száma nincs rögzítve, tartalmazza az 1, 2, … feladatokat is.

ütemezési információ azt mondja meg, hogy milyen ütemezést illető megkötések vannak a halmaz elemeire:

- $prec$ – létezik egy irányított a–ciklikus gráf amely meghatározza, hogy a munka elkezdéséhez mely munkák kellet már befejeződjeneik.
- r_j – egy–egy munkának tudjuk, hogy mely időponttól áll rendelkezésünkre.
- p_j – tudjuk, hogy egy–egy munka elvégzése mennyi időt vesz igénybe (ez minden adott).

optimalizálási kritérium arra vonatkozik, hogy a célfüggvényben mire fektetünk hangsúlyt:

- $C_{\max} = \max(C_j)$ – utolsó munka befejezási ideje legyen minél kisebb (gyorsan befejezni a csomagot).
- $\sum C_j = \sum \frac{C_j}{n}$ – a munka átlagos befejezási ideje legyen a lehető legkisebb (átlagosan keveset várunk a gépekre).

19.1. $1\|C_{max}$

Egy gépre teljes átfutási idő szerint optimizálunk. $C_{max} = \sum_{i=1}^n P_i \Rightarrow$ tehát csak felrakjuk a munkákat sorba, ügyelve arra, hogy ne létezzen pillanat amikor a gép áll és optimális megoldást kapunk.

19.2. $1|prec|C_{max}$

Most olyan sorrendben végezzük el a felrakást, hogy ne legyen munka amire a függőség még nem fejeződött be. Ez a sorrend egy irányított a-ciklikus gráfban (DAG) nem más mint a topologikus sorrend. Tehát első pontnak vegyünk egy forrás pontot a precedencia gráfból, ezt töröljük és ismételjük e folyamatot amíg létezik pont a gráfban. Ez polinomiális időben optimális megoldást ad.

Topologikus sorrend meghatározása

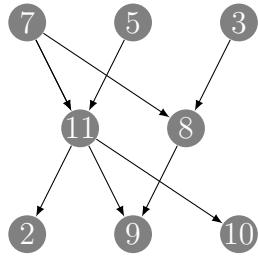
Adjunk hozzá a bemeneti DAG-hoz egy extra s csúcsot amelyet a végén törölni fogunk. minden $s \rightarrow v, v \in V$ élt rakjuk be a gráfba. minden pontra két címkét vezessünk: az egyik azt mutatja, hogy a pont elért e, és ha igen mely csúcsból; a másik érték meg azt jelzi, hogy a csúcs átvizsgált már e vagy sem. Kezdetben s elérte, de nem átvizsgált.

Amig létezik elérte, de még nem átvizsgált él addig ismételjük:

- kiválaszt a legkésőbb elérte de nem átvizsgált pontot $\rightarrow u$
- ha létezik $u \rightarrow v$ él a gráfban, ahol v nem elérte legyen v címkeje elérte u -ból.
- ha nem talál akkor u címkeje lesz átvizsgált, feljegyezve, hogy az algoritmus éppen hányadik lépésnél tart.

Az átvizsgálási sorrend lesz a topologikus sorrend, ha s nem tagja az eredeti gráfnak töröljük azt. Ha megfelelő adatstruktúrát használunk minden az élszám-mal arányos időt vesz igénybe.

Az itt látható gráfnak több topologikus rendezése is van:



- 7, 5, 3, 11, 8, 2, 10, 9
- 7, 5, 11, 2, 3, 10, 8, 9
- 3, 7, 8, 5, 11, 10, 9, 2
- 3, 5, 7, 11, 10, 2, 8, 9

19.3. $1\| \sum C_j$

Shortest–Processing–Time – SPT minél rövidebb egy munka annál korábban rakjuk fel.

A Shortest–Processing–Time optimális megoldást ad a feladatra.

A bizonyításhoz, tegyük fel, hogy az SPT nem ad optimális megoldást. Legyen p_1, p_2, \dots, p_n egy optimális megoldás, ami nem SPT típusú. Tehát létezik $p_i > p_{i+1}$ (szomszédos elemek).

Látszik, hogy az összegen ekkor csak C_i és C_{i+1} változik meg. Tehát a felcserélések végre hajtásával a $\sum C_j$ csökken, de ez ellentmondás a feltevésünknek, tehát az SPT optimális.

19.4. $P_2\|C_{max}$

Ez egy NP–nehéz feladatt mivel visszavezethető a partíció problémára. Legyen $a_1, a_2, \dots, a_n \in \mathbb{Z}^*$. A kérdés, hogy létezik-e $I \subseteq \{1, 2, \dots, n\}$ amire $\sum_{i \in I} a_i = \sum_{i \notin I} a_i$. Ez meg a részösszeg speciális esete. Vegyük most a megmunkálási időket $\{p_1, p_2, \dots, p_n\}$, ekkor, ha $\text{OPT} = \frac{\sum P_i}{2} \Leftrightarrow$ létezik p szám sorozatnak jó partíció.

Tehát nem létezik hatékony algoritmus a feladatra, ezért a problémának speciális eseteit és approximációs megoldásokat vizsgálunk meg.

19.5. $P\|C_{max}$

List Scheduling – LS – listás ütemezés Legyen k gép és $P_{i=1,n}$ feladat. Rakjuk fel ezeket sorrendben a gépekre, ha valamely gép végez megkapja a listában a következő munkát.

Ez egy elég primitív módszer hiszen gyakorlatilag csak arra ügyelünk, hogy a gépeink ne álljanak. Ennek ellenére, ha m gép van az LS egy $2 - \frac{1}{m}$ approximációt ad. Ennek belátásához legyen C_{max}^* az optimális ütemezés. Erre fent állnak a következő alsó becslések:

$$\frac{\sum P_i}{m} \leq C_{max}^* \quad \text{és} \quad P_{k \in i} \leq \max(P_i) \leq C_{max}^*.$$

Most legyen J_k az a munka melyet az LS ütemezés utoljára befejez. Ez egy t pillanatban kezdődik el és tart P_k idő intervallumot ($t + p_k = C_{max}$). E adott t időpontig az összes gép dolgozik, megállás nélkül, mert, ha ez nem így lenne stratégiánk szerint a P_k feladatot az a gép kapná meg. Tehát t pillanatig a P halmaz teljesen le van fedve (m darab gép, t időn keresztül):

$$m \cdot t \leq \sum_{i \neq k} P_i \Rightarrow t \leq \frac{\sum_{i \neq k} P_i}{m}$$

$$C_{\max} = t + P_k \leq \sum_{i \neq k} \frac{P_i}{m} + P_k = \sum_{i \neq k} \frac{P_i}{m} + \left(\frac{1}{m} P_k - \frac{1}{m} P_k \right) + P_k = \frac{\sum P_i}{m} + \left(1 - \frac{1}{m} \right) \cdot P_k$$

Erre meg használjuk fel az alsó becsléseink:

$$C_{\max} \leq C_{\max}^* + \left(1 - \frac{1}{m} \right) \cdot P_k \leq C_{\max}^* + \left(1 - \frac{1}{m} \right) C_{\max}^* = C_{\max}^* \cdot \left(2 - \frac{1}{m} \right)$$

Látható a bizonyításból, hogy a P_k mérete döntően befolyásolja az ütemezés hosszát, így ha ez kicsi javíthatunk az approximációs faktoron.

Longest Procesisng Time – LPT Rakjuk a munkákat csökkenő sorrendbe, és e új lista szerint ütemezünk (a legrövidebb marad a végire).

Az LPT ütemezés approximációs faktora $\left(\frac{4}{3} - \frac{1}{3m} \right)$. E jó approximációs közelítés annak köszönhető, hogy a gépek egyforma gyorsak, tudásuk azonos.

19.6. $P|prec|C_{\max}$ – Graham

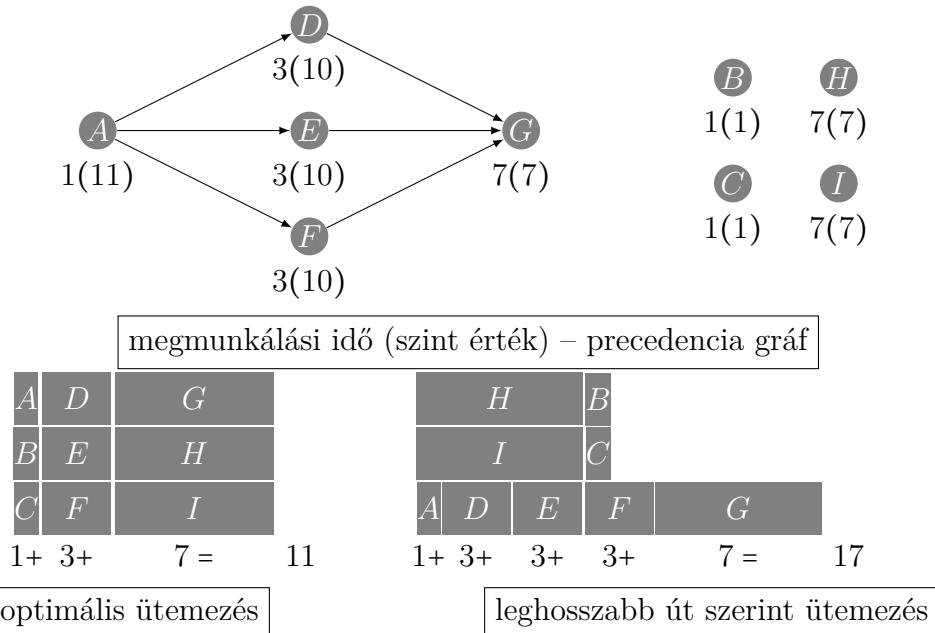
Rendezzük listába a munkákat. Egy gép megkap egy munkát, ha áll és a következő munka elkezdhető. Ez egy $2 - \frac{1}{m}$ approximáció. Ezen való javítás kulcsa, hogy hiába fejezzük be az utolsó feladathoz közeli feladatot mert a távoliak is be kell érjék ezeket (feltéve, hogy ezek függnek egymástól).

Legyen D irányított gráf a precedencia mátrixa a munkáknak. Ha bármely irányított úton összeadjuk az utakat alkotó pontok megfelelő megmunkálási időket, akkor alsó becslést kapunk egy optimális megmunkálási időre.

Tehát egy pont szintje legyen a pontból kiinduló irányított utak mentén vett megmunkálási idők összegének maximuma. Ezt megkaphatjuk, ha topologikus sorrendbe állítjuk a munkákat, majd felfelé lépve számoljuk a szint értékeit. Rendezzük a munkákat a szintük szerint csökkenő sorrendbe, és e lista szerint ütemezünk.

Jobb közelítést ez sem add, viszont ha az élszámunk magas jó eredményt szokott adni. Példa arra, hogy a leghosszabb út szerinti sorrend nem minden optimális, akkár az optimális kétszeresét is kiadhatja.

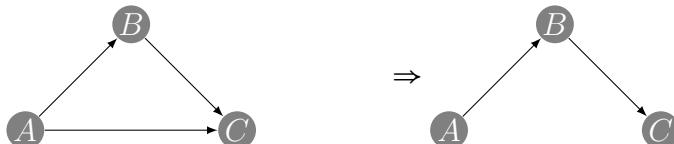
Legyen a 17 ábrán látható példa 9 munkával, 3 gépen. Ez nem $\frac{3}{2}$ approximáció mert az optimális és kapott arány $\frac{17}{11} \approx 1.54 > \frac{3}{2}$ (több gépes példán, vagy növelve G, H, I munkák értékét jobb ellenpélda is adható – azaz nagyobb eltérés $\frac{3}{2}$ -től).



17. ábra: A leghosszabb út szerinti sorrend nem minden optimális – példa

19.7. $P|prec, p_i = 1|C_{max}$

Ebben az alakban a feladat NP–nehéz. Ha $m = 2$ gép áll rendelkezésünkre a *Coffman-Graham* algoritmus optimális megoldást ad. D precedencia gráfon végezzük el a tranzitív redukciót:



Majd bontsuk csoportokra a csúcsokat: az első szinten azon csoportban találhatók akik kifoka 0, a másodikon akik kifoka 0, ha elhagyjuk az első szinten levő elemeket és így tovább. Majd az első szinten számozzuk meg a pontokat 1-től k_1 -ig, csoporton belül a sorrend nem számít.

A következő szinteken már a pontokat két körben számozzuk:

- első körben úgy számizzuk, hogy felsoroljuk mely pontokhoz van élük csökkenő sorrendben. Ez számozásokat lexikografikusan növekvő sorrendbe rendezése adja a csoporton belüli sorrendet.
- második körben a csoportbeli pontokat számizzuk át, folytatva a korábbi csoport számozásától a kapott csoportbeli sorrenddel.

A kapott számozás szerint csökkenő sorrendbe rakjuk fel a gépekre a munkákat mint egy listás ütemezés. Ez optimális megoldást ad.

19.8. $P|in_tree, p_j = 1|C_{max}$ – **Hu algoritmusa**

Ha D precedencia gráf egy s gyökerű F -befenyő⁴, akkor a szint szerinti ütemezés (gyökértől levő úthossz csökkenő sorrendben) optimális megoldást ad.

⁴A befenyő egy olyan irányított gyökeres fa, aminek az élei a gyökér felé vannak irányítva.