Kombinatorikus optimalizálás jegyzet

– az előadás és a kiadott szakirodalom alapján –

Készítette: *Schmidt Péter*

Alk. Mat., II. évf.

2010-2011

TARTALOM

mailto: szabos@ttk.pte.hu

Kombinatorikus optimalizálás

Egész értékű problémák megoldásához gyakran felhasználhatók a feladatok kombinatorikus tulajdonságai, amelyek esetenként az általános egészértékű programozási módszereknél hatékonyabb eljárások felépítését teszik lehetővé. A kombinatorikus optimalizálás témakörén belül ezért módszertani megközelítésben tárgyaljuk az általános egész értékű problémák és a speciálisan kombinatorikai (főképpen gráfelméleti) jellegű optimalizálási feladatok megoldhatóságát és megoldását.

A kombinatorikai problémák alapkérdései:

Egy véges struktúrának adott tulajdonságú része vagy elrendezése

1) létezik-e;

2) hány van;

3) melyik a legjobb;

4) hogyan található meg a leggyorsabban?

Halmazok

Halmaz lefedése

Pl. órarend összeállítása, személyzet beosztása járatokhoz, stb.

Feladat

Egy alaphalmaz bizonyos részhalmazaihoz költségeket rendelünk.

Fedjük le az alaphalmazt a lehető legkisebb költséggel!

Példa

Feladat

Alaphalmaz: $U = \{1,2,3,4,5,6,7\}$

Fedőhalmazok és költségeik:

$A_1 = \{1,3,5\}$ $c_1 = 4$

$A_2 = \{1,4,5,6\}$ $c_2 = 5$

$A_3 = \{4,5,6\}$ $c_3 = 5$

$A_4 = \{1,7\}$ $\qquad c_4 = 4$

$A_5 = \{2,6,7\}$ $\qquad c_5 = 3$

$A_6 = \{3,5,7\}$ $\qquad c_6 = 3$

LP-modell

Az alaphalmaz elemeinek és a fedőhalmazoknak az incidencia-mátrixa

U	1	2	3	4	5	6	7
A_1	1		1		1		
A_2	1			1	1	1	
A_3				1	1	1	
A_4	1						1
A_5		1				1	1
A_6			1		1		1

Vezessük be a következő bináris változókat:

$$x_i = \begin{cases} 1, \text{ ha } A_i \text{ része a fedésnek} \\ 0, \text{ egyébként} \end{cases}$$

A fedési feladat lehetséges megoldásainak ki kell elégíteniük azt a feltételt, hogy az alaphalmaz minden egyes eleme le legyen fedve. Ezeket a feltételeket az incidencia-mátrix transzponáltja segítségével lehet szemléletesen felírni:

U	x_1	x_2	x_3	x_4	x_5	x_6	
1	1	1		1			≥ 1
2					1		≥ 1
3	1					1	≥ 1
4		1	1				≥ 1
5	1	1	1			1	≥ 1
6		1	1		1		≥ 1
7				1	1	1	≥ 1

Vegyük észre, hogy a második feltétel baloldalán csak egyetlen változó szerepel, ami azt jelenti, hogy a változóhoz tartozó fedőhalmaz minden lehetséges megoldásnak eleme lesz, mert az alaphalmaz második elemét egyedül ő fedi.

A legkisebb költség elérése érdekében a $\sum_{i=1}^{6} c_i x_i$ célfüggvényt minimalizálni kell.

A halmazfedési feladathoz tartozó LP-modell tehát a következő:

$$
\begin{cases}
4x_1 + 5x_2 + 5x_3 + 4x_4 + 3x_5 + 3x_6 \to \min \\
------------------------ \\
x_1 + x_2 + x_4 \geq 1 \\
x_5 \geq 1 \\
x_1 + x_6 \geq 1 \\
x_2 + x_3 \geq 1 \\
x_1 + x_2 + x_3 + x_6 \geq 1 \\
x_2 + x_3 + x_5 \geq 1 \\
x_4 + x_5 + x_6 \geq 1 \\
------------------------ \\
x_1, x_2, x_3, x_4, x_5, x_6 \in \{0,1\}
\end{cases}
$$

Sperner-rendszerek

A Sperner-rendszerek speciális tulajdonságú extremális halmazrendszerek, amelyeken értelmezett egy irreflexív, szimmetrikus reláció.

Legyen alaphalmazunk egy tetszőleges, n elemű véges halmaz hatványhalmaza. Az alaphalmaz elemeinek az általánosság megszorítása nélkül megfeleltethetjük az őket meghatározó kiválasztási függvényeket vagy karakterisztikus függvényeket, ezáltal halmazok helyett az n dimenziós Hamming-tér vektoraival dolgozhatunk.

Részben rendezés

Értelmezzük az n dimenziós Hamming-tér vektorain a kanonikus részben rendezést: $\mathbf{x} \leq \mathbf{y}$ akkor és csak akkor, ha a reláció a vektorok megfelelő komponenseire páronként fennáll.

Összehasonlíthatatlanság

Definíció szerint $\mathbf{x}$ és $\mathbf{y}$ összehasonlíthatatlanok, ha sem $\mathbf{x} \leq \mathbf{y}$, sem $\mathbf{y} \leq \mathbf{x}$ nem áll fenn.

Megjegyzések

Az összehasonlíthatatlanság egy irreflexív, szimmetrikus reláció.

A halmazok nyelvére visszavetítve az összehasonlíthatatlanság azt jelenti, hogy egyik halmaz sem tartalmazza a másikat, más szóval a Hasse-diagramban nincs az egyikből a másikba vezető út.

Sperner-rendszer

Az *n* dimenziós Hamming-tér egy *K* része Sperner-rendszer, ha abban bármely két elem összehasonlíthatatlan.

Megjegyzés

Az üres halmaz és az egyelemű halmazok megállapodás szerint Sperner-rendszerek.

Példa

$K = \{0011, 1100, 0110, 1001\}$ egy Sperner-rendszer.

Sperner-lemma

Ha $K \subseteq \{0,1\}^n$ egy Sperner-rendszer, akkor $|K| \le \binom{n}{\left\lfloor \frac{n}{2} \right\rfloor}$

Bizonyítás

A lemmánál erősebb állítást bizonyítunk, nevezetesen azt, hogy ha $K \subseteq \{0,1\}^n$ egy Sperner-rendszer, akkor $\sum_{\mathbf{x} \in K} \frac{1}{\binom{n}{|\mathbf{x}|}} \le 1$, ahol $|\mathbf{x}|$ az $\mathbf{x}$ vektor Hamming-súlya.

Ugyanis $\binom{n}{|\mathbf{x}|}$ akkor maximális, ha $|\mathbf{x}| = \left\lfloor \frac{n}{2} \right\rfloor$, tehát az előző összeg alulról becsülhető a következőképpen: $|K| \cdot \frac{1}{\binom{n}{\left\lfloor \frac{n}{2} \right\rfloor}} \le \sum_{\mathbf{x} \in K} \frac{1}{\binom{n}{|\mathbf{x}|}} \le 1$, ebből pedig átszorzással következik a Sperner-lemma.

Az erősebb állítás bizonyításához tekintsünk egy tetszőleges maximális utat a $\{0,1\}^n$ tér Hasse-diagramjának az aljától a tetejéig:

$$\mathbf{0} = \mathbf{y}_0 \le \mathbf{y}_1 \le \cdots \le \mathbf{y}_n = \mathbf{1}$$

Vegyük észre a következőket:

1) $|\mathbf{y}_i| = i$

2) $n!$ számú különböző maximális út létezik.

3) Rögzített $\mathbf{x} \in K$ esetén $|\mathbf{x}|! \cdot (n - |\mathbf{x}|)!$ olyan maximális út létezik, amelyen $\mathbf{x}$ rajta van.

4) Tetszőleges $\mathbf{u}, \mathbf{v} \in K$ esetén nem lehet a Sperner-rendszer mindkét eleme rajta egy maximális úton, hiszen az út pontjai összehasonlíthatók.

A fentiekből következik, hogy minden maximális út a Sperner-rendszernek legfeljebb egy elemét tartalmazhatja, és a Sperner-rendszer egy $\mathbf{x} \in K$ eleme összesen $|\mathbf{x}|! \cdot (n - |\mathbf{x}|)!$ különböző úton szerepelhet. A Sperner-rendszer összes elemére tehát felírható, hogy $\sum_{\mathbf{x} \in K} |\mathbf{x}|! \cdot (n - |\mathbf{x}|)! \leq n!$, amiből az erősebb állítás következik.

Megjegyzés

A Stirling-formula felhasználásával belátható, hogy $\lim_{n \to \infty} \binom{n}{\left\lfloor \frac{n}{2} \right\rfloor} \cdot \frac{1}{2^n} = 0$, tehát a Sperner-rendszerek méretnövekedése exponenciális alatt marad.

Telített Sperner-rendszer

Egy Sperner-rendszer **telített**, ha ahhoz a Hamming-térnek már nem lehet több elemét hozzáadni.

Tétel

Egy $2n$ elemű halmaz részhalmazaiból álló maximális méretű Sperner-rendszer elemei az n elemű részhalmazok.

Tétel

Legyen $K = \left\{ \mathbf{x} \in \{0,1\}^n \mid \mathbf{A}\mathbf{x} = \mathbf{b} \right\}$ és $H = \left\{ \mathbf{x} \in \{0,1\}^n \mid \mathbf{A}\mathbf{x} = \mathbf{0} \right\}$.

Ha $H = \{\mathbf{0}\}$, akkor K egy Sperner-rendszer.

Bizonyítás

Indirekt úton, tegyük fel, hogy $H = \{\mathbf{0}\}$ és K nem Sperner-rendszer, azaz léteznek $\mathbf{u}, \mathbf{v} \in K$, amelyek összehasonlíthatók, például $\mathbf{u} \geq \mathbf{v}$.

K definíciója miatt $\mathbf{A}\mathbf{u} = \mathbf{A}\mathbf{v} = \mathbf{b}$, ezért $\mathbf{A}(\mathbf{u} - \mathbf{v}) = \mathbf{0}$, tehát $\mathbf{0} \neq \mathbf{u} - \mathbf{v} \in H$, ami ellentmond a $H = \{\mathbf{0}\}$ feltevésnek.

Következmény

Egzakt fedési probléma lehetséges megoldásai Sperner-rendszerek.

Egzakt fedés

Tétel

Az n dimenziós Hamming-tér tetszőleges $L \subseteq \{0{,}1\}^n$ részéhez található olyan $\mathbf{A}$ mátrix és $\mathbf{b}$ vektor, hogy $L = \{\mathbf{x} \in \{0{,}1\}^n \mid \mathbf{Ax} \le \mathbf{b}\}$

Bizonyítás

Az L-hez tartozó $\mathbf{A}$ mátrixot és $\mathbf{b}$ vektort úgy kapjuk meg, hogy minden L-en kívül eső $\mathbf{z} \in \{0{,}1\}^n, \mathbf{z} \notin L$ vektorra felírjuk azt az egyenlőtlenséget, amely rajta kívül a Hamilton-tér összes többi elemére fennáll, a következőképpen:

Legyen $I = \{ i \mid z_i = 1\}$ a $\mathbf{z}$ vektor 1 komponenseinek az indexhalmaza, és $J = \{ j \mid z_j = 0\}$ a $\mathbf{z}$ vektor 0 komponenseinek az indexhalmaza.

A vektorok komponenseire vonatkozó következő egyenlőtlenség a $\mathbf{z}$ vektoron kívül minden más vektorra igaz:

$$\sum_{i \in I} x_i - \sum_{j \in J} x_j \le |I| - 1$$

Példa

Legyen $L = \{0001, 0010, 0011, 0101, 0110, 1001, 1010, 1011, 1100, 1110, 1111\}$

Ekkor $\overline{L} = \{0000, 0100, 0111, 1000, 1101\}$

Az $\overline{L}$ elemeihez tartozó egyenlőtlenségek rendre a következők:

$$\begin{cases} 0000: & -x_1 - x_2 - x_3 - x_4 \le -1 \\ 0100: & x_1 - x_2 + x_3 + x_4 \le 0 \\ 0111: & -x_1 + x_2 + x_3 + x_4 \le 2 \\ 1000: & x_1 - x_2 - x_3 - x_4 \le 0 \\ 1101: & x_1 + x_2 - x_3 + x_4 \le 2 \end{cases}$$

A fenti egyenlőtlenségrendszerből közvetlenül leolvasható az L-hez tartozó

$$\mathbf{A} = \begin{pmatrix} -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & 1 \\ -1 & 1 & 1 & 1 \\ 1 & -1 & -1 & -1 \\ 1 & 1 & -1 & 1 \end{pmatrix} \text{ és } \mathbf{b} = \begin{pmatrix} -1 \\ 0 \\ 2 \\ 0 \\ 2 \end{pmatrix}.$$

Az $\mathbf{A}$ mátrixot és $\mathbf{b}$ vektort kis gyakorlattal közvetlenül az $\overline{L}$ elemeiről is leolvashatjuk.

Megjegyzés

A fenti tétel tartalmi jelentése, hogy az egyenlőtlenségfeltételes bináris optimalizálási problémák bizonyos értelemben teljesen strukturálatlanok.

A tétel kiegészítő párja Bradley tétele, amely szerint az egyenlőségfeltételes bináris optimalizálási problémák bizonyos értelemben tökéletesen strukturáltak és meghatározottak.

Egészértékű függvény

Az $f:\mathbf{R}^n \to \mathbf{R}$ függvény **egészértékű** a $H \subseteq \mathbf{R}^n$ halmazon, ha annak minden elemére egész értéket vesz fel: $\forall \mathbf{x} \in H:\ f(\mathbf{x}) \in \mathbf{Z}$

Bradley-lemma

Legyen f és g két egészértékű függvény a $H \subseteq \mathbf{R}^n$ halmazon, ekkor léteznek olyan u és w egész számok, amelyekre $Z(u,w) = \{\mathbf{x} \in H : u \cdot f(\mathbf{x}) + w \cdot g(\mathbf{x}) = 0\}$ a két függvény közös zérushelyeinek a halmaza.

Bizonyítás

Legyen a két függvény közös zérushelyeinek a halmaza $R = \{\mathbf{x} \in H:\ f(\mathbf{x}) = g(\mathbf{x}) = 0\}$

$R \subseteq Z(u,w)$ nyilvánvalóan tetszőleges (u,w) számokra fennáll.

Azt kell bizonyítanunk, hogy léteznek olyan (u,w) egész számok, amelyekre $Z(u,w) \subseteq R$.

Az általánosság megszorítása nélkül feltehető, hogy u és w relatív prímek $(u, w \neq 0)$.

Adott (u,w) számokhoz vezessük be a következő nívóhalmazokat:

- $K_1 = \{\mathbf{x} \in H : \operatorname{sgn}(w) \cdot f(\mathbf{x}) \leq -|w|\}$
- $K_2 = \{\mathbf{x} \in H : \operatorname{sgn}(w) \cdot f(\mathbf{x}) \geq |w|\}$
- $K_3 = \{\mathbf{x} \in H : \operatorname{sgn}(u) \cdot g(\mathbf{x}) \leq -|u|\}$
- $K_4 = \{\mathbf{x} \in H : \operatorname{sgn}(u) \cdot g(\mathbf{x}) \geq |u|\}$

A nívóhalmazok segítségével definiáljuk a következő korlátokat:

- $I_1 = \inf_{\mathbf{x} \in K_1} g(\mathbf{x})$ $\quad S_1 = \sup_{\mathbf{x} \in K_1} g(\mathbf{x})$
- $I_2 = \inf_{\mathbf{x} \in K_2} g(\mathbf{x})$ $\quad S_2 = \sup_{\mathbf{x} \in K_2} g(\mathbf{x})$
- $I_3 = \inf_{\mathbf{x} \in K_3} f(\mathbf{x})$ $\quad S_3 = \sup_{\mathbf{x} \in K_3} f(\mathbf{x})$
- $I_4 = \inf_{\mathbf{x} \in K_4} f(\mathbf{x})$ $\quad S_4 = \sup_{\mathbf{x} \in K_4} f(\mathbf{x})$

A korlátok alapján tekintsük a következő feltételhalmazokat:

- $A = \{u < I_1, S_1 < u, -w < I_4, S_4 < -w\}$
- $B = \{-u < I_2, S_2 < -u, w < I_3, S_3 < w\}$

Tegyük fel mármost, hogy $\mathbf{y} \in Z(u, w)$, tehát $u \cdot f(\mathbf{y}) = -w \cdot g(\mathbf{y}) = t$.

Mivel u és w relatív prímek, így $t = u \cdot w \cdot r$, azaz $f(\mathbf{y}) = wr$ és $g(\mathbf{y}) = -ur$.

Esetvizsgálattal megmutatható, hogy az A feltételhalmaz bármely eleméből következik $r \geq 0$, és a B feltételhalmaz bármely eleméből következik $r \leq 0$.

(Például $u < I_1$ esetén, indirekt módon, tegyük fel, hogy $r < 0$, így $f(\mathbf{y}) = -w, -2w, \ldots$ értékeket vehet fel.

Vizsgáljuk az $I_1 = \inf_{\mathbf{x} \in K_1} g(\mathbf{x})$ kiszámíthatóságát:

- Ha $w < 0$, akkor a feltevés miatt $f(\mathbf{y}) \geq -w = |w|$, másrészt $\operatorname{sgn}(w) = -1$, így $\operatorname{sgn}(w) \cdot f(\mathbf{y}) \leq -|w|$.
- Ha $w < 0$, akkor a feltevés miatt $f(\mathbf{y}) \leq -w = -|w|$, másrészt $\operatorname{sgn}(w) = 1$, így $\operatorname{sgn}(w) \cdot f(\mathbf{y}) \leq -|w|$.

Tehát w bármely értéke kiszámíthatóvá teszi $I_1 = \inf_{\mathbf{x} \in K_1} g(\mathbf{x})$-t, így tehát a feltevés szerint bármely w-re $u < I_1 = \inf_{\mathbf{x} \in K_1} g(\mathbf{x}) \leq g(\mathbf{y}) = -ur$ kell, hogy teljesüljön. Azonban $u < 0$ esetén az $u < -ur$ egyenlőtlenségből $r \geq 0$ adódik, ami ellentmond az eredeti feltevésnek, következésképpen az $u < I_1$ feltételből következik $r \geq 0$.)

Megmutattuk tehát, hogy ha az A és B feltételhalmazok egy-egy eleme teljesül, akkor abból $r = 0$ következik, ami éppen azt jelenti, hogy $Z(u,w) \subseteq R$, vagyis $Z(u,w) = R$.

Bradley tétele

Egy $\mathbf{Ax} = \mathbf{b}$ egészegyütthatós egyenlőségfeltételes bináris optimalizálási feladat feltételrendszere mindig redukálható egyetlen alkalmasan megválasztott egyenletre.

Bizonyítás

Ha az egyenlőségfeltételek egyetlen egyenletből állnak, akkor készen vagyunk. Máskülönben az egyenlőségfeltételek közül válasszunk ki kettőt tetszőlegesen, és a segítségükkel képezzük a következő egészértékű függvényeket:

$$\begin{cases} \mathbf{a}_i \cdot \mathbf{x} = b_i \Rightarrow f(\mathbf{x}) = \mathbf{a}_i \cdot \mathbf{x} - b_i \\ \mathbf{a}_j \cdot \mathbf{x} = b_j \Rightarrow g(\mathbf{x}) = \mathbf{a}_j \cdot \mathbf{x} - b_j \end{cases}$$

Legyen $u = 1$ és $w = \max\limits_{\mathbf{x} \in \{0,1\}^n} \left(| f(\mathbf{x}) | + 1 \right) > 0$, ekkor $-(w-1) \leq f(\mathbf{x}) \leq (w-1)$.

Vegyük észre, hogy a Bradley-lemmában meghatározott feltételek közül ekkor $S_1 < u$ és $-u < I_2$ egyaránt fennáll, következésképpen a két függvény közös zérushelyeinek a halmaza éppen $Z(u,w) = \{ \mathbf{x} \in H : u \cdot f(\mathbf{x}) + w \cdot g(\mathbf{x}) = 0 \}$, vagyis a kiválasztott két feltétel helyettesíthető az egyetlen, $u \cdot f(\mathbf{x}) + w \cdot g(\mathbf{x}) = 0$ feltétellel, ami szintén egészegyütthatós egyenlőségfeltétel.

Az egyenlőségfeltételek halmaza így lépésenként redukálható, végeredményben egyetlen feltételre.

Következmény

Bármely egzakt fedési problémához létezik olyan hátizsák feladat, amelynek a megoldása az egzakt fedési problémának is megoldása lesz.

Párosítások

A **párosítás** egy gráf éleinek olyan részhalmaza, amelyben két élnek nincs közös pontja. Egy párosítás **teljes**, ha a gráf összes csúcspontját lefedi. Egy gráf **páros**, ha a csúcspontok két osztályba sorolhatók úgy, hogy a gráf bármely élének a két végpontja különböző osztályban van. Egy páros gráf **teljes**, ha a csúcspontok két osztályának bármely pontpárja éllel van összekötve.

Maximális élszámú párosítás keresése

Feladat

Egy egyszerű gráfban

1) Hány elemű a maximális élszámú párosítás?

2) Hogyan lehet a leggyorsabban maximális élszámú párosítást találni?

3) Hány maximális élszámú párosítás létezik?

4) Soroljuk fel az összes maximális élszámú párosítást!

Modell

Legyen $\Gamma = (V, E)$ egyszerű gráf, $|V| = n$.

Vezessük be a következő bináris változókat:

$$x_{ij} = \begin{cases} 1, \text{ ha az } (i,\, j) \text{ él létezik és a párosítás eleme} \\ 0 \text{ egyébként} \end{cases}$$

A párosítás élszáma: $\sum_{i=1}^{n} \sum_{j=1}^{n} x_{ij}$

Mivel párosítást keresünk, az egy pontra illeszkedő élek közül legfeljebb egy kerülhet a párosításba:

- $\forall i \in V : \sum_{j=1}^{n} x_{ij} \leq 1$
- $\forall j \in V : \sum_{i=1}^{n} x_{ij} \leq 1$

Az optimalizálási probléma modellje:

$$\begin{cases} \sum_{i=1}^{n} \sum_{j=1}^{n} x_{ij} \to \max \\ \text{-------------} \\ \forall i \in V : \sum_{j=1}^{n} x_{ij} \leq 1 \\ \forall j \in V : \sum_{i=1}^{n} x_{ij} \leq 1 \\ \text{-------------} \\ x_{ij} \in \{0,1\} \end{cases}$$

Módszerek

Maximális élszámú párosítás keresése páros gráfban: az alternáló utak módszere

Induljunk ki a gráf egy tetszőleges párosításából (például egyetlen élből), és keressünk ennél jobb párosítást, azaz javítsuk a megoldást, amennyiben ez lehetséges.

1. Keressünk az aktuális párosításra nézve *szabad csúcsot*, amelyre az aktuális párosítás egyetlen éle sem illeszkedik.

2. Építsük fel a szabad csúcsból a lehető leghosszabb *alternáló utat*, amelynek pontosan minden második éle eleme az aktuális párosításnak.

3. Ha a leghosszabb alternáló út páratlan hosszú, akkor ez egy *javító út* az aktuális párosításra nézve, mert az útnak az aktuális párosításhoz tartozó éleit a párosításból elhagyva, a maradék éleket pedig a párosításhoz hozzávéve, eggyel nagyobb élszámú párosítást kapunk.

Berge tétele (1957): Egy párosítás élszáma akkor és csak akkor maximális, ha rá nézve nem létezik a gráfban javító út.

Maximális élszámú párosítás keresése egyszerű gráfban: az alternáló fák módszere, Edmonds algoritmusa

Berge tétele tetszőleges egyszerű gráfra igaz. A kérdés: hogyan keressünk a gráfban javító utat, illetve hogyan győződjünk meg róla, ha ilyen nem létezik.

Javító utat úgy keresünk a gráfban, hogy egy *alternáló fa* építésével feltárjuk egy adott pontból a lehetséges alternáló utakat, és így vagy találunk javító utat, vagy kimerítjük a gráfot, amivel igazoljuk, hogy az aktuális párosításra nézve javító út nincs, tehát az aktuális párosítás maximális.

0. Minden csúcspont és minden él címkézetlen.

1. Induljunk ki egy szabad csúcspontból. (Ha ilyen nincs, akkor az aktuális párosítás teljes, így maximális.) A szabad csúcsot jelöljük meg a *'külső'* címkével. Legyen ez az aktuális alternáló fa gyökere.

2. Keressünk címkézetlen élt, amely illeszkedik az aktuális alternáló fa egy *'külső'*, *u* csúcsára. (Ha ilyen nincs, akkor nem létezik az aktuális párosításra nézve javító út, tehát az aktuális párosítás maximális).

- Ha az él másik, v végpontja címkézetlen, akkor az (u,v) él címkéje legyen *'igen'*. Ha v szabad csúcs, akkor az aktuális alternáló fa gyökeréből a v-hez vezető alternáló út egy javító út az aktuális párosításra nézve; máskülönben létezik az aktuális párosításnak egy (v,t) éle: a (v,t) él címkéje is legyen *'igen'*, a v csúcs címkéje legyen *'belső'*, a t csúcs címkéje legyen *'külső'*, és ismételjük meg a 2. lépést.
- Ha az él másik, v végpontja *'belső'*, akkor a (v,t) él címkéje is legyen *'nem'*, és ismételjük meg a 2. lépést. (Ennek az élnek a hozzávétele páros kör kialakulását eredményezné az aktuális alternáló fában.)
- Ha az él másik, v végpontja *'külső'*, akkor álljunk meg. (Ennek az élnek a hozzávétele páratlan kör kialakulását eredményezné az aktuális alternáló fában.)

Az alternáló fa az *'igen'* címkéjű élekből áll.

/ld. *Edmonds algoritmus*, Imreh-Imreh: 40. old/

Minimális súlyú teljes párosítás keresése páros gráfban

Feladat

Egyenlő méretű osztályokból álló, élsúlyozott páros gráfban

1) Van-e teljes párosítás?

2) Mi a teljes párosítások minimális súlya?

3) Adjunk meg egy minimális súlyú teljes párosítást!

4) Hány minimális súlyú teljes párosítás van?

5) Soroljuk fel a minimális súlyú teljes párosításokat!

Modell

Legyen $\Gamma = (A \cup B, E)$ páros gráf $w : E \to \mathbf{R}$ élsúlyokkal.

A gráf osztályainak a mérete legyen $n = |A| = |B|$.

Feltehető, hogy az élsúlyok nemnegatív számok: $w(i, j) = w_{ij} \geq 0$

Vezessük be a következő bináris változókat:

$$x_{ij} = \begin{cases} 1, \text{ ha az } (i,\ j) \text{ él a párosítás eleme} \\ 0 \text{ egyébként} \end{cases}$$

A párosítás súlya: $\sum_{(i,j)\in E} w_{ij} \cdot x_{ij}$

Mivel teljes párosítást keresünk, az egy pontra illeszkedő élek közül pontosan egy kerülhet a párosításba:

- $\forall i \in A: \sum_{j \in B} x_{ij} = 1$
- $\forall j \in B: \sum_{i \in A} x_{ij} = 1$

Az optimalizálási probléma modellje:

$$\begin{cases} \sum_{(i,j) \in E} w_{ij} \cdot x_{ij} \to \min \\ \text{--------------} \\ \forall i \in A: \sum_{j \in B} x_{ij} = 1 \\ \forall j \in B: \sum_{i \in A} x_{ij} = 1 \\ \text{--------------} \\ x_{ij} \in \{0,1\} \end{cases}$$

Módosított modell: hozzárendelési feladat (*Assignment Problem*)

Tegyük teljessé a páros gráfot a hiányzó élek hozzávételével úgy, hogy a hiányzó élek súlyát „végtelenül nagyra” választjuk. A gyakorlatban elegendő, ha a „végtelenül nagy” súly a megadott súlyok összegénél nagyobb szám:

- Legyen $W > \sum_{(i,j) \in E} w_{ij}$
- Legyen $c: A \times B \to \mathbf{R},\ c(i,j) = c_{ij} = \begin{cases} w_{ij}, \text{ha } (i,j) \in E \\ W \text{ egyébként} \end{cases}$

A teljessé tett gráfra vonatkozó lineáris optimalizálási probléma az $\mathbf{A}[c_{ij}]$ **hozzárendelési feladat**:

$$\begin{cases} \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} \cdot x_{ij} \to \min \\ \text{--------------} \\ \forall i: \sum_{j=1}^{n} x_{ij} = 1 \\ \forall j: \sum_{i=1}^{n} x_{ij} = 1 \\ \text{--------------} \\ x_{ij} \in \{0,1\} \end{cases}$$

A kapcsolat az eredeti és a módosított feladat között a következő: az eredeti feladatnak akkor és csak akkor van megoldása, ha a módosított feladat optimális megoldása kisebb, mint *W*, és ebben az esetben a módosított feladat optimális megoldása egyben optimális megoldása az eredeti feladatnak is.

A módosított modell elemzése

A hozzárendelési feladat olyan **speciális szállítási feladat**, ahol minimális költségű teljes párosítást keresünk egy élsúlyozott páros gráfban.

A hozzárendelési feladat lehetséges megoldása egy olyan bináris mátrix, amelynek minden sorában és minden oszlopában pontosan egy darab 1 elem szerepel. Megfordítva: minden $n \times n$-es bináris mátrix, amelynek minden sorában és oszlopában pontosan egy darab 1 elem szerepel, lehetséges megoldása a feladatnak. Az ilyen mátrixok, azaz megoldások száma $n!$.

Rögzített n mellett a hozzárendelési feladatot egyértelműen meghatározza a *súlymátrix* vagy *költségmátrix*.

Megoldási módszerek

Magyar módszer

Hozzárendelési feladatra a disztribúciós módszer nem hatékony.

A **magyar módszer** a duál feladat megoldását célozza a költségmátrix redukciójával.

Duál feladat

$$\max w = \sum_i u_i + \sum_j v_j$$

$$u_i + v_j \le c_{ij}$$

$$u_i, v_j \in \mathbf{R}$$

A költségmátrix redukciója

Vegyük észre, hogy a u_i illetve v_j duál változók értékei a $\mathbf{C} = (c_{ij})$ költségmátrixról közvetlenül leolvashatók: u_i rendre az *i*-edik sor legkisebb eleme, v_j pedig a *j*-edik oszlop legkisebb eleme!

Redukáljuk a költségmátrixot a következőképpen:

1. Csökkentsük minden sor elemeinek értékét a sorban található minimális elem (u_i) értékével.

2. Csökkentsük minden oszlop elemeinek értékét az oszlopban található minimális elem (v_j) értékével.

Komplementaritási tétel

A primál feladat **x** lehetséges megoldása és a duál feladat **u,v** lehetséges megoldása akkor és csak akkor optimális, ha $x_{ij} > 0$ esetén $u_i + v_j = c_{ij}$.

Vegyük észre, hogy a redukált költségmátrix elemei éppen a $c_{ij} - u_i - v_j$ értékek, a redukált költségmátrixban tehát az összes duál változó értéke 0.

Legyen $x_{ij} = 1$ ott, ahol $c_{ij} - u_i - v_j = 0$, mindenütt másutt pedig $x_{ij} = 0$.

A költségmátrix lefedése

1. Válasszuk ki a költségmátrixból a lehető legtöbb független 0 elemet.

Kőnig Dénes tétele, hogy egy páros gráfban a maximális párosítás éleinek a száma megegyezik a gráf éleit lefogó csúcspontok minimális számával.

Kőnig tétele alapján a redukált költségmátrixban a független nullák száma megegyezik az összes nullát lefedő sor- és oszlop-vonalak minimális számával.

2. Húzzuk meg a redukált költségmátrix 0 elemeit lefedő sor- és oszlopvonalakat.

A duál megoldás javítása

1. Keressük meg a legkisebb fedetlen elemet, legyen ez δ.

2. A fedetlen elemekből vonjuk ki δ-t.

3. Az egyszeresen fedett elemeket hagyjuk változatlanul.

4. A kétszeresen fedett elemekhez adjunk hozzá δ-t.

Megjegyzés

A fenti transzformációval ekvivalens megfogalmazás, hogy

1. A fedetlen sorok minden eleméből vonjunk ki δ-t.

2. A lefedett oszlopok minden eleméhez adjunk hozzá δ-t.

További ekvivalens megfogalmazást kaphatunk, ha felcseréljük a *sor* és *oszlop* szavakat.

Eredmény

Ha a lefedett sorok száma eredetileg I_s, a lefedett oszlopok száma pedig eredetileg I_o, akkor a fenti transzformációval a $w = \sum_i u_i + \sum_j v_j$ célfüggvény $(n - I_s)\delta$ mennyiséggel csökkent, és $I_o\delta$ mennyiséggel nőtt, összességében tehát a célfüggvény növekménye: $d = (n - I_s)\delta - I_o\delta = (n - I_s - I_o)\delta$

Amennyiben az eredeti párosításunk nem volt teljes, úgy $(n - I_s - I_o) > 0$, tehát a javítással a célfüggvény értéke nőtt.

Az eljárás vége

Ha a redukált, majd rekurzívan javított költségmátrixból kiválasztott maximális számú független 0 elemmel a mátrix minden sorát és oszlopát le tudjuk fedni, akkor az optimális megoldás a kiválasztott 0 elemeknek megfelelő éleket tartalmazó teljes párosítás.

Megjegyzés

A kiválasztási feltételeken szokás a következőképpen lazítani: $x_{ij} \in [0,1]$

Példák

Munkák optimális kiosztása

Feladat

Meghatározott számú munkát kell kiosztani ugyanennyi dolgozónak úgy, hogy a munkavégzés összköltsége (pl. a ráfordított idő) a lehető legkisebb legyen.

1) Kiosztható-e minden munka, ha egy dolgozó csak egy munkát végezhet el?

2) Mekkora a teljes munkavégzés minimális összköltsége?

3) Adjuk meg a munkák egy optimális kiosztását.

4) Hány különböző optimális kiosztás létezik?

5) Soroljuk fel az összes optimális munkamegosztást!

Modell

A dolgozók illetve az elvégzendő munkák számát jelölje *n*.

A dolgozókat *i*-vel, az elvégzendő munkákat *j*-vel indexeljük. $(1 \leq i \leq n,\ 1 \leq j \leq n)$

Jelölje w_{ij} annak a költségét, hogy az *i* dolgozó végzi el a *j* munkát. Mivel nem garantált, hogy minden dolgozó minden egyes munkát képes elvégezni, vagyis a dolgozókat és az általuk elvégezhető munkákat összekötve nem szükségképpen jutunk teljes páros gráfhoz, így a w_{ij} költségmátrixnak lehetnek hiányzó elemei, és lehetséges, hogy a feladatnak egyáltalán nincs megoldása.

Vezessük be a következő bináris változókat:

$$x_{ij} = \begin{cases} 1, \text{ ha az } i \text{ dolgozó kapja a } j \text{ munkát} \\ 0 \text{ egyébként} \end{cases}$$

Amennyiben az *i* dolgozó nem végezheti el a *j* munkát, tehát a w_{ij} költség nem létezik, abban az esetben garantáltan $x_{ij} = 0$.

A munkavégzés összköltsége: $\sum_{(i,j)\in E} w_{ij} \cdot x_{ij}$

A munkák kiosztásának feltételei:

- Minden dolgozó pontosan egy munkát kaphat: $\forall i : \sum_{j=1}^{n} x_{ij} = 1$
- Minden munka pontosan egy dolgozóhoz kerülhet: $\forall j : \sum_{i=1}^{n} x_{ij} = 1$

Az optimalizálási modell:

$$\begin{cases} \sum_{(i,j)\in E} w_{ij} \cdot x_{ij} \to \min \\ -------------- \\ \forall i : \sum_{j=1}^{n} x_{ij} = 1 \\ \forall j : \sum_{i=1}^{n} x_{ij} = 1 \\ -------------- \\ x_{ij} \in \{0,1\} \end{cases}$$

Módosított modell: hozzárendelési feladat

Ahhoz, hogy lineáris programozással is kezelhető, azaz teljes költségmátrixú feladatot tudjunk megoldani, azt az esetet, amikor egy dolgozó nem képes egy adott munkát végrehajtani (vagy valamilyen más okból azt nem kaphatja meg) úgy kezeljük, hogy a vonatkozó költséget „végtelenül nagyra” választjuk, ami a gyakorlatban elegendő, ha a megadott költségek összegénél egy nagyobb szám: $W > \sum_{(i,j)\in E} w_{ij}$. Ezáltal az eredeti feladat módosított változatához jutunk.

A módosított (kiegészített) optimalizálási modell:

$$
\begin{cases}
\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}\cdot x_{ij} \to \min \\
-------------- \\
\forall i: \sum_{j=1}^{n} x_{ij} = 1 \\
\forall j: \sum_{i=1}^{n} x_{ij} = 1 \\
-------------- \\
x_{ij} \in \{0,1\}
\end{cases}
$$

A kapcsolat az eredeti és a módosított feladat között a következő: az eredeti feladatnak akkor és csak akkor van megoldása, ha a módosított feladat optimális megoldása kisebb, mint W, és ebben az esetben a módosított feladat optimális megoldása egyben optimális megoldása az eredeti feladatnak is.

Párválasztás

Feladat

Meghatározott számú leány mindegyike tetszési sorrendet állít fel azonos számú fiúról. Egy lehetséges kapcsolat boldogság értékét a leánynak a fiúra vonatkozó tetszési indexe mutatja. Adjuk meg azt a teljes párosítást, amely a legnagyobb összboldogságot eredményezi.

1) Mekkora az elérhető legnagyobb összboldogság?

2) Adjuk meg a maximális összboldogságot eredményező párválasztást!

3) Hány különböző optimális párválasztás létezik?

4) Soroljuk fel az optimális párosításokat!

Modell

A leányok illetve fiúk számát jelölje n. A leányokat i-vel, a fiúkat j-vel indexeljük.

Jelölje c_{ij} az i leánynak a j fiúra vonatkozó tetszési indexét. Feltehető, hogy a tetszési index nemnegatív $\left(c_{ij} \geq 0\right)$.

Vezessük be a következő bináris változókat:

$$x_{ij} = \begin{cases} 1, \text{ha az } i \text{ leány párja a } j \text{ fiú} \\ 0 \text{ egyébként} \end{cases}$$

Az összboldogság: $\sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} \cdot x_{ij}$

A párválasztás feltételei:

- Minden leány pontosan egy fiút választhat: $\forall i : \sum_{j=1}^{n} x_{ij} = 1$
- Minden fiút pontosan egy leány választhat: $\forall j : \sum_{i=1}^{n} x_{ij} = 1$

Az optimalizálási modell az $\mathbf{A}[c_{ij}]$ hozzárendelési feladat:

$$\begin{cases} \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} \cdot x_{ij} \to \max \\ ------------- \\ \forall i : \sum_{j=1}^{n} x_{ij} = 1 \\ \forall j : \sum_{i=1}^{n} x_{ij} = 1 \\ ------------- \\ x_{ij} \in \{0,1\} \end{cases}$$

Minimális súlyú teljes párosítás keresése tetszőleges gráfban

Feladat

Páros csúcsú élsúlyozott gráfban

1) Van-e teljes párosítás?

2) Mi a teljes párosítások minimális súlya?

3) Adjunk meg egy minimális súlyú teljes párosítást!

4) Hány minimális súlyú teljes párosítás van?

5) Soroljuk fel a minimális súlyú teljes párosításokat!

Modell

Legyen $\Gamma = (V, E)$ páros csúcsú gráf $w : E \to \mathbf{R}$ élsúlyokkal.

A gráf csúcspontjainak a száma legyen $|V| = 2n$.

Feltehető, hogy az élsúlyok nemnegatív számok: $w(i, j) = w_{ij} \geq 0$

Vezessük be a következő bináris változókat:

$$x_{ij} = \begin{cases} 1, \text{ ha az } (i, j) \text{ él a párosítás eleme} \\ 0 \text{ egyébként} \end{cases}$$

A párosítás súlya: $\sum\limits_{(i,j)\in E} w_{ij} \cdot x_{ij}$

Mivel teljes párosítást keresünk, az egy pontra illeszkedő élek közül pontosan egy kerülhet a párosításba:

- $\forall i \in A : \sum\limits_{j\in B} x_{ij} = 1$
- $\forall j \in B : \sum\limits_{i\in A} x_{ij} = 1$

Az optimalizálási probléma modellje:

$$\begin{cases} \sum\limits_{(i,j)\in E} w_{ij} \cdot x_{ij} \to \min \\ -------------- \\ \forall i \in A : \sum\limits_{j\in B} x_{ij} = 1 \\ \forall j \in B : \sum\limits_{i\in A} x_{ij} = 1 \\ -------------- \\ x_{ij} \in \{0,1\} \end{cases}$$

Módosított modell: hozzárendelési feladat (*Assignment Problem*)

Tegyük teljessé a páros gráfot a hiányzó élek hozzávételével úgy, hogy a hiányzó élek súlyát „végtelenül nagyra” választjuk. A gyakorlatban elegendő, ha a „végtelenül nagy” súly a megadott súlyok összegénél nagyobb szám:

- Legyen $W > \sum\limits_{(i,j)\in E} w_{ij}$
- Legyen $c : A \times B \to \mathbf{R},\ c(i, j) = c_{ij} = \begin{cases} w_{ij}, \text{ ha } (i,j) \in E \\ W \text{ egyébként} \end{cases}$

A teljessé tett gráfra vonatkozó lineáris optimalizálási probléma az $\mathbf{A}[c_{ij}]$ **hozzárendelési feladat**:

$$\begin{cases} \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}\cdot x_{ij} \to \min \\ -------------- \\ \forall i: \sum_{j=1}^{n} x_{ij} = 1 \\ \forall j: \sum_{i=1}^{n} x_{ij} = 1 \\ -------------- \\ x_{ij} \in \{0,1\} \end{cases}$$

A kapcsolat az eredeti és a módosított feladat között a következő: az eredeti feladatnak akkor és csak akkor van megoldása, ha a módosított feladat optimális megoldása kisebb, mint *W*, és ebben az esetben a módosított feladat optimális megoldása egyben optimális megoldása az eredeti feladatnak is.

A módosított modell elemzése

A hozzárendelési feladat olyan **speciális szállítási feladat**, ahol minimális költségű teljes párosítást keresünk egy élsúlyozott páros gráfban.

A hozzárendelési feladat lehetséges megoldása egy olyan bináris mátrix, amelynek minden sorában és minden oszlopában pontosan egy darab 1 elem szerepel. Megfordítva: minden $n \times n$-es bináris mátrix, amelynek minden sorában és oszlopában pontosan egy darab 1 elem szerepel, lehetséges megoldása a feladatnak. Az ilyen mátrixok, azaz megoldások száma $n!$.

Rögzített n mellett a hozzárendelési feladatot egyértelműen meghatározza a *súlymátrix* vagy *költségmátrix*.

Példák

Raktár felszámolása

Feladat

Egy kiürítendő raktárból $2n$ számú konténert kell párosával elszállítani. A konténerek páronkénti elszállítási költsége adott.

1) Kiüríthető-e a raktár?

2) Mekkora a raktár kiürítésének minimális költsége?

3) Adjuk meg a raktárfelszámolás egy optimális stratégiáját.

4) Hány különböző optimális stratégia létezik?

5) Soroljuk fel az összes optimális stratégiát!

Klikkek

Klikk-keresési problémák modellezése 0-1 programozási feladatként

Probléma

Adott egy Γ (általában egyszerű) gráf és egy $k \in \mathbf{N}$ természetes szám.

1. a) Döntsük el, hogy van-e a gráfban *k* méretű klikk.

1. b) Mutassunk meg egy *k* méretű klikket, ha van!

1. c) Hány különböző *k* méretű klikk van a gráfban?

1. d) Soroljuk fel az összes *k* méretű klikket!

2. a) Mekkora a gráfban a legnagyobb méretű klikk?

2. b) Mutassunk meg egy legnagyobb méretű klikket!

2. c) Hány különböző legnagyobb méretű klikk van a gráfban?

2. d) Soroljuk fel az összes legnagyobb méretű klikket!

Példa

Feladat

Legyen a Γ gráf adjacencia-mátrixa a következő:

Γ	1	2	3	4	5	6
1		1		1	1	
2	1		1			
3		1		1		1
4	1		1		1	
5	1			1		
6			1			

Mekkora a gráfban található legnagyobb méretű klikk?

LP-modell

Vezessük be a következő bináris változókat:

$$x_i = \begin{cases} 1, \text{ha az } i \text{ csúcs tagja a klikknek} \\ 0, \text{egyébként} \end{cases}$$

A $\sum_{i=1}^{6} x_i$ célfüggvényt maximalizálni kell.

A feltételek megfogalmazásához a gráf ún. anti-incidencia mátrixát (vagyis a gráf komplementerének incidencia-mátrixát) érdemes felírni, amelyben az éllel össze nem kötött csúcsokat (a „be nem húzott éleket”) jelenítjük be:

x_1	x_2	x_3	x_4	x_5	x_6	
1		1				≤ 1
1					1	≤ 1
	1		1			≤ 1
	1			1		≤ 1
	1				1	≤ 1
		1		1		≤ 1
			1		1	≤ 1
				1	1	≤ 1

Mivel a klikkben nem jelenhetnek meg éllel össze nem kötött csúcsok, ezért az anti-incidencia mátrix egy sorában megjelölt csúcsok közül legfeljebb az egyik lehet tagja a klikknek.

Megjegyzés

Mivel bináris változókat vizsgálunk, így a **valós relaxálás** gyorsan adhat **nemleges** választ egy *k* méretű klikk keresésére.

Kombinatorikus vágások mohó színezéssel

Vágásokat azért alkalmazunk, hogy csökkentsük a probléma méretét. NP-teljes probléma esetén a méretcsökkentés érdekében pl. szuboptimális, de hatékony mohó algoritmusok is kiválóan alkalmasak lehetnek arra, hogy vágásokat generáljunk.

Klikk-keresési problémák esetén kombinatorikus alapon is vágást generálhatunk pl. gráfszínezéssel. A minimális gráfszínezés maga is NP-teljes feladat, de a kombinatorikus vágáshoz nem szükséges optimális színezést megadni: elegendő a pontokat mohón, az első lehetséges színnel kiszínezni.

A gráfban nyilván nem lehet nagyobb klikk, mint ahány színosztály van. Minden színosztályra megfogalmazható egy olyan pótlólagos feltétel, hogy a színosztály elemei közül legfeljebb az egyik lehet a klikk tagja. Ez a **kombinatorikus vágás**. Azon színosztályok szerinti kombinatorikus vágások, amelyek 2-nél több csúcsot tartalmaznak, egymagukban is több eredeti feltételt helyettesítenek, így csökkentik a feltételek számát.

A fenti feladatban például a mohó színezés az alábbi eredményhez vezet:

1. szín: 1, 3

2. szín: 2, 4, 6

3. szín: 5

A második színosztályhoz tartozó kombinatorikus vágás:

$$x_2 + x_4 + x_6 \leq 1$$

Ez a vágás önmagában helyettesíti a harmadik, ötödik és hetedik feltételt, és a valós relaxáltról levágja az $x_i = \frac{1}{2}$ alakú megoldásokat.

Megjegyzés

További vágásokat találhatunk, ha a Γ gráf komplementerében köröket keresünk. Ezek a körök lánc-egyenlőtlenségeket jelentenek az eredeti feltételhalmazban, amelyeket összeadva egy újabb vágáshoz jutunk.

A példafeladatban egy kör:

$$\begin{aligned} x_1 + x_2 &\leq 1 \\ x_2 + x_3 &\leq 1 \\ x_3 + x_4 &\leq 1 \\ x_4 + x_5 &\leq 1 \\ x_5 + x_6 &\leq 1 \\ x_6 + x_1 &\leq 1 \end{aligned}$$

Az ehhez tartozó vágás:

$$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \leq 3$$

Ez a vágás csak akkor teljesülhet egyenlőtlenségre, ha a körön éppen minden második változó 1-es. Könnyű ellenőrizni, hogy ez a megoldás megengedett-e, és ha nem, akkor a vágás jobb oldalán álló kapacitási korlát még eggyel csökkenthető.

Maximális klikk méretének felső és alsó becslése mohó színezéssel

Feladat

Keressük meg a legnagyobb méretű klikket az alábbi adjacencia-mátrixszal leírható gráfban:

×		●	●		●	●	●		●	●	
	×			●		●				●	
●		×	●	●		●		●		●	●
●		●	×				●				●
	●	●		×		●	●		●		
●					×		●	●		●	●
●	●	●		●		×		●	●	●	
●			●	●	●		×	●			●
		●			●	●	●	×		●	●
●				●		●			×		
●	●	●			●	●		●		×	●
		●	●		●		●	●		●	×

Megoldás

Első lépésként vessük alá a fenti gráfot egy gyors inspekciónak: számoljuk ki és tüntessük fel a gráf sorai mellett az egyes csomópontok fokszámát:

×		●	●		●	●	●		●	●		7
	×			●		●				●		3
●		×	●	●		●		●		●	●	7
●		●	×				●				●	4
	●	●		×		●	●		●			5
●					×		●	●		●	●	5
●	●	●		●		×		●	●	●		7
●			●	●	●		×	●			●	6
		●			●	●	●	×		●	●	6
●				●		●			×			3
●	●	●			●	●		●		×	●	7
		●	●		●		●	●		●	×	6

Mivel a legnagyobb fokszám 7, így a gráfban legfeljebb 8-as méretű klikk lehet.

Mivel azonban 7-es fokszámú pontból csak 4 van, így a gráfban legfeljebb 7-es méretű klikk lehet.

7-es méretű klikkhez legalább 6 fokú pontok kellenek. A legalább 6 fokú pontok száma éppen 7, ezek azonban nem alkotnak klikket (pl. hiányzik az 1—12 él).

A fokszámok vizsgálata alapján tehát gyorsan megállapítható, hogy a gráfban legfeljebb 6-os méretű klikk található.

Színezzük ki mohó módon a gráfot:

1	2	3	4	5
1	3	4	7	12
2	6	5	8	
9	10	11		

Mohó módon 5 színnel sikerült a gráfot kiszínezni, ami azt jelenti, hogy a gráfban legfeljebb 5-ös méretű klikket találhatunk.

Mivel az utolsó színosztálynak csak egyetlen pontja van, ez azt jelenti, hogy amennyiben van a gráfban 5-ös méretű klikk, úgy annak ez a pont biztosan az eleme lesz. Folytassuk a mohó színezést rekurzív módon, hogy megtaláljuk a 12-es csomópont szomszédjai által kifeszített részgráfban a legnagyobb klikket. Írjuk tehát egymás alá az előző színezés színosztályait, és a megadott sorrendben válogassuk ki az utolsó (legkisebb méretű) színosztály utolsó elemének szomszédjait:

1
2
9
3
6
10
4
5
11
7
8
12

Az utolsó színosztály utolsó elemének szomszédjai a fenti sorrendben: 9, 3, 6, 4, 11, 8

Színezzük a kapott részgráfot mohó módon:

1	2	3
9	3	11
4	6	8

Írjuk a kapott színosztályok elemeit függőlegesen az előző mellé, és folytassuk az eljárás egészen addig, amíg egypontú részgráfhoz nem jutunk:

1	9	9	9
2	4	4	
9	3	6	
3	6		
6	11		
10	8		
4			
5			
11			
7			
8			
12			

A fenti oszlopok utolsó elemei klikket alkotnak a gráfban, van tehát a gráfban legalább 4 méretű klikk: {12, 8, 6, 9}

A teljes gráf mohó színezésével tehát felső korlátot, majd a mohó színezés rekurzív alkalmazásával alsó korlátot is találtunk a legnagyobb méretű klikkre.

Megjegyzés

A mohó színezés eredménye többféle előkészítéssel (prekondicionálással) javítható, például úgy is, hogy a csomópontokat a színezés előtt fokszám szerint csökkenő sorrendbe állítjuk.

Monoton mátrixok generálása klikk-kereséssel

Definíció

Egy **monoton mátrix** olyan $n \times n$-es számtáblázat, amelyre a következő feltételek teljesülnek:

- *cella feltétel*: a táblázat cellái a $0,1,\ldots,n \in \mathbf{N}$ értékek egyikét tartalmazzák (a 0 értéket a cella üresen hagyásával jelezzük, és a többi feltétel ellenőrzése során figyelmen kívül hagyjuk);
- *sor feltétel*: a sorok (balról jobbra) szigorúan monoton növők;
- *oszlop feltétel*: az oszlopok (lentről felfelé) szigorúan monoton növők;
- *meredekségi feltétel*: bármely két azonos pozitív számot tartalmazó cellát összekötő egyenes meredeksége pozitív.

Példák

1

	1
1	

	2
1	

	2	3
2		
1		

	1	3
3		
1		2

Megjegyzés

A monoton mátrixok elméletének alapproblémája az, hogy adott mérethez megtaláljuk a lehető legjobban kitöltött (legtöbb pozitív cellát tartalmazó) monoton mátrixokat. A maximálisan kitöltött monoton mátrixokat **optimálisnak** nevezzük. A definíció alapján a következő állítások egyszerűen beláthatók:

- Optimális monoton mátrix első sora és oszlopa pontosan egy darab 1-est tartalmaz, ami eltolható a bal alsó sarokba.
- Optimális monoton mátrix utolsó sora és oszlopa pontosan egy darab *n*-est tartalmaz, ami eltolható a jobb felső sarokba.

Monoton mátrixok gráf-modellje és az alapprobléma LP-modellje

Tekintsük azt az n^3 pontból álló gráfot, amelynek pontjait olyan pozitív egész (x, y, z) számhármasokkal azonosítjuk, ahol $1 \le x, y, z \le n$. Egy (x, y, z) csomópontot úgy interpretálunk, hogy az $n \times n$-es számtáblázat x sorának és y oszlopának kereszteződésében található cella tartalma z.

Kössük össze azokat a csomópontokat, amelyek kielégítik a monoton mátrixra vonatkozó feltételeket. Az (x_1, y_1, z_1) és (x_2, y_2, z_2) csomópontok tehát akkor és csak akkor vannak a gráfban éllel összekötve, ha

- $x_1 < x_2 \Rightarrow z_1 < z_2$ (sor feltétel)
- $y_1 < y_2 \Rightarrow z_1 < z_2$ (oszlop feltétel)
- $z_1 = z_2 \Rightarrow \dfrac{y_2 - y_1}{x_2 - x_1} > 0$ (meredekségi feltétel)

A fenti módon definiált gráfban minden klikk egy $n \times n$-es monoton mátrixot definiál. Az $n \times n$-es alapprobléma tehát ekvivalens a maximális klikk megkeresésének problémájával egy n^3 méretű gráfban, ami az előzőek alapján megfelel egy egész értékű lineáris programozási feladatnak.

Pakolási problémák modellezése

Pakolási problémának nevezzük egy halmaz lehető legteljesebb lefedését előre megadott részhalmazainak valamely részrendszerével. A pakolási problémával ekvivalens klikk-keresési probléma alapgráfjának pontjait az előre megadott részhalmazok képezik, amelyek közül a diszjunktak vannak éllel összekötve. Ezen a gráfon minden klikk egy lehetséges pakolásnak felel meg. Lényeges azonban, hogy nem a legnagyobb *méretű* klikket keressük a gráfban. A gráf pontjai ugyanis a nekik megfelelő részhalmazok méretével súlyozottak, és a feladat a legnagyobb *súlyú* klikk megkeresése. Abban a speciális esetben persze, ha az eredeti halmazt azonos méretű részekkel kell maximálisan lefedni, azaz a gráf pontjainak a súlya azonos, a maximális méretű klikkek lesznek egyben maximális súlyúak is.

Martin Gardner egy nevezetes feladata 1971-ből

Kérdés

Elhelyezhető-e 42 darab 1×2×4-es méretű tégla egy 7×7×7-es méretű kockában?

A probléma elemzése

Martin Gardner problémája egy olyan pakolási feladat, amely azonos méretű részhalmazokkal célozza egy alaphalmaz maximális fedését. Vizsgáljuk meg a problémát kombinatorikai szemszögből. Egy kis téglát 6 féleképpen állíthatunk be a nagy kockába úgy, hogy a tégla bal alsó első sarka a kocka megadott pozíciójára essen. Összesen tehát 7×7×7×6 módon lehet egy kis téglát a nagy kockában elhelyezni, beleszámítva azokat a szélső helyzeteket is, amikor a tégla a kocka jobb, felső vagy hátsó lapján már kilóg – ennél tehát kevesebb tényleges pakolási lehetőségünk van egyetlen téglára nézve.

Mivel egy kis tégla térfogata 8 egység, így a nagy kockába legfeljebb $\left\lfloor \frac{7^3}{8} \right\rfloor = 42$ darab kis tégla helyezhető el. A kérdés tehát az, hogy ez a maximális pakolás megvalósítható-e.

A probléma modellezése 1-0 programozási feladattal

Írjuk a nagy kocka pozícióinak és a kis tégla elhelyezési lehetőségeinek incidencia mátrixát. Ahhoz, hogy az LP-modell feltételrendszere a mátrixról könnyen leolvasható legyen, a 7×7×7 pozíciót a sorokban, a (kevesebb, mint) 7×7×7×6 elhelyezési lehetőséget pedig az oszlopokban tüntessük fel:

$$\begin{array}{ccc} & 343\times 6 > m \text{ db bináris változó} & \\ 343\,\text{feltétel} & \boxed{\begin{array}{c}\\ \\ \end{array}} & \begin{array}{c}\le 1\\ \vdots \\ \le 1\end{array}\end{array}$$

Ha tehát eleve töröljük a lehetséges 343×6 közül azokat az irreleváns oszlopokat, ahol kevesebb, mint 8 db 1-es szerepel, a pakolási feladat pontos feltételrendszerét kapjuk. Az eredeti kérdés arra vonatkozik, hogy a $\sum_i x_i$ célfüggvény maximuma az adott feltételek mellett eléri-e a 42-t.

A probléma modellezése klikk-kereséssel

Tekintsük azt a gráfot, amelynek a csomópontjai az előző LP-feladat változói, és a gráf két pontja akkor és csak akkor legyen éllel összekötve, ha a megfelelő két változó oszlopvektorának skaláris szorzata 0 (azaz a változókkal reprezentált téglák nem „lógnak” egymásba). Keressük ebben a gráfban a maximális klikket. Az eredeti kérdés arra vonatkozik, hogy a gráf maximális klikkmérete eléri-e a 42-t.

A probléma modellezése egzakt fedéssel

Vegyük észre, hogy a nagy kocka bármely (külső vagy belső) lapja páratlan számú kis kockából áll, egy kis tégla azonban a lapot csak páros számú (2, 4 vagy 8) kis kockában metszheti. Ez azt jelenti, hogy ha létezik az eredeti pakolási problémának megoldása, akkor az a nagy kocka minden lapján pontosan egy kis kockát hagy fedetlenül. A fedetlen kis kockák elhelyezkedési lehetőségeinek a száma $(7!)^2$.

Ha a lefedetlen kiskockáktól eltekintünk, vagyis kihúzzuk a fenti incidencia mátrixból azokat a oszlopokat (0-ra állítjuk azokat a változókat), amelyek a lefedetlen kis kockákba belemetszenek, akkor a maradék incidencia mátrix feltételi rendszerét teljesen kiélesíthetjük, és a pakolási feladat éles változatához, egy egzakt fedési problémához jutunk.

Az eredeti kérdés arra vonatkozik, hogy a $(7!)^2$ darab egzakt fedési probléma közül létezik-e legalább egynek megoldása.

Klikk-keresési problémák modellezése kvadratikus optimalizálási feladatként

Egy klikk-keresési probléma általános 0-1 programozási modellje a következőképpen formalizálható:

$$\begin{cases} \sum_{i=1}^{n} x_i \to \max \\ \exists i, j : x_i + x_j \leq 1 \\ x_i \in \{0,1\} \end{cases}$$

A fenti absztrakt modellben a gráf pontjainak a száma *n*, és a feltételek pontosan azokra a csomópontokra állnak, amelyek nincsenek éllel összekötve (tehát nem lehetnek egy klikkben).

Feltétel nélküli kvadratikus optimalizálási modell

Vegyük észre, hogy a feltételekben szereplő bináris lineáris egyenlőtlenségeknek létezik egy ekvivalens, kvadratikus egyenlőség alakja:

$$x_i + x_j \leq 1 \Leftrightarrow x_i x_j = 0$$

Ezen alapul a következő ötlet: fogalmazzuk át a feltételeket büntetési tételekké, és építsük bele őket a célfüggvénybe. Az optimumkeresés a büntetések minimalizálásán alapul majd, tehát az eredeti célfüggvény tételeit is negálni kell ahhoz, hogy minimumfeladathoz jussunk. Az új célfüggvény, és egyben a klikk-keresés feltétel nélküli kvadratikus optimalizálási modellje tehát a következő lesz:

$$\sum_{\substack{i<j \\ (i,j)\notin E}} x_i x_j - \sum_{i=1}^{n} x_i \to \min$$

Ha tiszta kvadratikus alakokat szeretnénk a célfüggvényben látni, akkor a változók bináris voltára való tekintettel ez is megtehető, hiszen egy bináris változó pozitív egész hatványainak értéke azonos, így a célfüggvény az alábbi formában is írható:

$$\sum_{\substack{i<j \\ (i,j)\notin E}} x_i x_j - \sum_{i=1}^{n} x_i^2 \to \min$$

A tiszta kvadratikus alak előnye, hogy lehetővé teszi az indexelés alkalmazása helyett a mátrix-szorzásra való átalakítást, és ezzel a célfüggvény egyszerűbb, áttekinthetőbb felírását:

$$\sum_{\substack{i<j \\ (i,j)\notin E}} x_i x_j - \sum_{i=1}^{n} x_i^2 = \frac{1}{2}\mathbf{x}^{\mathrm{T}}\mathbf{B}\mathbf{x}$$

Mivel az eredeti lineáris modellben a feltételeket a gráf éllel össze nem kötött elemeire fogalmaztuk meg, és ezek a feltételek a **B** bináris mátrixba 1 értékű elemként kerültek bele, így a **B** főátlón kívüli 0 elemei éppen az eredeti gráf éleit reprezentálják. Más szóval a **B** bináris mátrix főátlón kívüli elemei megegyeznek az eredeti gráf komplementerének az adjacencia-mátrixában található értékekkel. Jelöljük az eredeti gráf adjacencia-mátrixát **A**-val $\left(a_{ij} = 1 \Leftrightarrow (i, j) \in E\right)$, a gráf komplementerének adjacencia-mátrixát pedig $\overline{\mathbf{A}}$-val ekkor a pontos összefüggés a mátrixok között a következő: $\mathbf{B} = \overline{\mathbf{A}} - 2\mathbf{I}$, ahol **I** a megfelelő méretű egységmátrix.

Tétel

Legyen **A** egy egyszerű gráf adjacencia-mátrixa. Ekkor a gráfban található maximális méretű klikket az

$$\mathbf{x}^{\mathrm{T}}\left(\overline{\mathbf{A}} - 2\mathbf{I}\right)\mathbf{x}$$

bináris kvadratikus alak minimumhely-vektorának pozitív komponensei jelölik ki.

Bizonyítás

A tételt megelőző gondolatsor alapján azt kell csak megmutatnunk, hogy a gráf (egy) maximális méretű klikkje által meghatározott $\boldsymbol{\alpha}$ vektor globális minimumhelye lesz az $q(\mathbf{x}) = \mathbf{x}^{\mathrm{T}}\left(\overline{\mathbf{A}} - 2\mathbf{I}\right)\mathbf{x} = \sum_{\substack{i<j \\ a_{ij}=0}} x_i x_j - \sum_{i=1}^{n} x_i$ kvadratikus alaknak.

Legyen tehát $\boldsymbol{\alpha}$ egy k méretű maximális klikk a gráfban. Ekkor

$$q(\boldsymbol{\alpha}) = \sum_{\substack{i<j \\ a_{ij}=0}} x_i x_j - \sum_{i=1}^{n} x_i = 0 - k = -k\ .$$

Tekintsünk most egy tetszőleges $\mathbf{x}$ vektort, amely egy részgráfot reprezentál. Megmutatjuk, hogy $q(\mathbf{x})$ értéke nem lehet kisebb, mint $-k$. Legyen ugyanis $\boldsymbol{\beta} \le \mathbf{x}$ a részgráfban található legnagyobb méretű klikk. Ekkor $q(\boldsymbol{\beta}) \ge q(\boldsymbol{\alpha}) = -k$.

Kiegészítve mármost a $\boldsymbol{\beta}$ klikket a részgráf további pontjaival, minden egyes pont hozzáadásával eggyel megnő ugyan a negatív előjelű $\sum_{i=1}^{n} x_i$ tag abszolút értéke, ám annyival nő ugyanakkor a pozitív $\sum_{\substack{i<j \\ a_{ij}=0}} x_i x_j$ tag értéke is, ahány ponttal nincs összekötve a részgráfban a hozzáadott pont. Mivel pedig $\boldsymbol{\beta}$ volt a legnagyobb klikk a részgráfban, a maradék pontok $\boldsymbol{\beta}$-nak legalább egy pontjával nincsenek összekötve, tehát a pozitív $\sum_{\substack{i<j \\ a_{ij}=0}} x_i x_j$ tag értéke legalább eggyel nő. Következésképpen, ha $\boldsymbol{\beta}$-t kiegészítjük az $\mathbf{x}$ részgráfra, eközben a kvadratikus alak értéke nem csökkenhet, így $q(\mathbf{x}) \geq q(\boldsymbol{\beta}) \geq q(\boldsymbol{\alpha}) = -k$, vagyis $\boldsymbol{\alpha}$ globális minimumhelye lesz a kvadratikus alaknak.

Feltételes kvadratikus optimalizálási modell

Az előzőekben láttuk, hogy a klikk-keresési probléma átírható egy feltétel nélküli kvadratikus optimalizálási feladatra. Annak természetesen semmi akadálya nincs, hogy az eredeti feltételrendszer tetszőleges tagjainak hozzáadásával feltételes optimalizálási problémává bővítsük a feladatot. Ennek célszerűsége akkor mutatkozik meg, ha a feltételek explicit figyelembe vétele megkönnyíti vagy meggyorsítja az optimalizálási probléma megoldását.

Az optimalizálási problémák bonyolultságát minőségi és mennyiségi jellemzők határozzák meg. A célfüggvényt és a feltételeket felépítő algebrai kifejezések jellege, továbbá a változók alaphalmaza (értelmezési tartománya) a probléma bonyolultságának minőségi jellemzőihez tartoznak, és alapvetően meghatározzák egy probléma megoldásának lehetséges módjait (módszereit, algoritmusait). A bonyolultság elsődleges mennyiségi jellemzőjét a probléma mérete, vagyis a változók és a feltételek száma jelenti, azonban a célfüggvényben és a feltételekben szereplő együtthatók puszta értéke, vagy egyszerűen a nemnulla együtthatók száma is jelentősen befolyásolhatja a megoldások meghatározásához szükséges számítási igényt. Ha például egy $\mathbf{x}^{\mathbf{T}}\mathbf{B}\mathbf{x}$ bináris kvadratikus alak mátrixa nagyon ritka (vagy éppen nagyon sűrű), az lerövidítheti a szélsőérték megtalálásának az idejét.

Megfontolandó tehát, hogy a kvadratikus átfogalmazás előtt pl. kombinatorikus módszerekkel (színezéssel, vágásokkal stb.) csökkentsük egy klikk-keresési probléma méretét, ritkítsuk mátrixát, a gráf bizonyos területeinek a kizárásával jobban behatároljuk a keresési teret, hogy aztán egy kiegészítő feltételekkel ellátott, de jelentősen redukált (ritka mátrixú) kvadratikus alak szélsőértékét kelljen csak meghatározni. Erre a hibrid módszerre mutatunk most példát a mohó színezés alkalmazásánál már látott feladat kapcsán.

Probléma

Keressük meg a legnagyobb méretű klikket az alábbi adjacencia-mátrixszal leírható gráfban:

×		•	•		•	•	•		•	•	
	×			•		•				•	
•		×	•	•		•		•		•	•
•		•	×				•				•
	•	•		×		•	•		•		
•					×		•	•		•	•
•	•	•		•		×		•	•	•	
•			•	•	•		×	•			•
		•			•	•	•	×		•	•
•				•		•			×		
•	•	•			•	•		•		×	•
		•	•		•		•	•		•	×

Kisméretű feltételes kvadratikus optimalizálási modell kialakítása

Tudjuk, hogy ha az eredeti gráf komplementerének élei alapján közvetlenül írjuk fel a 0-1 programozási feladat feltételrendszerét, akkor általában nagyszámú feltételhez jutunk, aminek a közvetlen átírása nem túl ritka mátrixú kvadratikus alakot eredményez.

Azt is láttuk, hogy a feltételek száma kombinatorikus vágásokkal csökkenthető, viszont az összevont feltételek már nem építhetők be a kvadratikus célfüggvénybe.

A két módszer célszerű ötvözésével viszont a nagyméretű LP-probléma helyett egy kisebb méretű feltételes kvadratikus optimalizálási problémához juthatunk úgy, hogy a színezéssel és kombinatorikus vágásokkal összevonható élfeltételeket összevonjuk, a maradékot pedig kvadratikus átalakítással beépítjük a célfüggvénybe!

Színezzük ki tehát mohó módon a gráfot, és a nagyméretű színosztályok alapján hajtsunk végre kombinatorikus vágást az eredeti LP-modell feltételrendszerén:

1	2	3	4	5
1	3	4	7	12
2	6	5	8	
9	10	11		

Tekintsük most nagynak a legalább 3 méretű színosztályokat, és a kapott vágásokkal kezdjük el felépíteni a kvadratikus modell feltételrendszerét:

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	
1	1							1				≤ 1
		1			1				1			≤ 1
			1	1						1		≤ 1

A kapott vágásokat *élként visszajelöljük* az eredeti adjacencia-mátrixba, hogy azonos módszerrel további vágásokhoz jussak:

×	○	●	●		●	●	●	○	●	●	
○	×			●		●		○		●	
●		×	●	●	○	●		●	○	●	●
●		●	×	○			●			○	●
	●	●	○	×		●	●		●	○	
●		○			×		●	●	○	●	●
●	●	●		●		×		●	●	●	
●			●	●	●		×	●			●
○	○	●			●	●	●	×		●	●
●		○		●	○	●			×		
●	●	●	○	○	●	●		●		×	●
		●	●		●		●	●		●	×

Újraszínezzük az élekkel bővített gráfot:

1	2	3	4	5
1	2	4	9	11
5	3	6	10	
12	8	7		

Megint találtunk 3 vágást, amit felveszünk a feltételrendszerbe…

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	
1	1							1				≤ 1
		1			1				1			≤ 1
			1	1						1		≤ 1
1				1							1	≤ 1
	1	1					1					≤ 1
			1		1	1						≤ 1

… és élként visszajelölünk az adjacencia-mátrixba is:

×	○	●	●	○	●	●	●	○	●	●	○
○	×	○		●		●	○	○		●	
●	○	×	●	●	○	●	○	●	○	●	●
●		●	×	○	○	○	●			○	●
○	●	●	○	×		●	●		●	○	○
●		○	○		×	○	●	●	○	●	●
●	●	●	○	●	○	×		●	●	●	
●	○	○	●	●	●		×	●			●
○	○	●			●	●	●	×		●	●
●		○		●	○	●			×		
●	●	●	○	○	●	●		●		×	●
○		●	●	○	●		●	●		●	×

Vegyük észre, hogy az 1-es és 3-as csomópontok telítetté váltak, így azok további vágásokban nem vehetnek részt.

Újraszínezzük az élekkel bővített gráf telítetlen pontjait:

1	2	3	4	5	6
2	5	7	9	11	12
4	6	8			
10					

Megint találtunk egy vágást, amit felvehetünk a feltételrendszerbe…

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	
1	1							1				≤ 1
		1			1				1			≤ 1
			1	1						1		≤ 1
1				1							1	≤ 1
	1	1					1					≤ 1
			1		1	1						≤ 1
	1		1						1			≤ 1

… és élként visszajelölünk az adjacencia-mátrixba is:

×	○	●	●	○	●	●	●	○	●	●	○
○	×	○	○	●		●	○	○	○	●	
●	○	×	●	●	○	●	○	●	○	●	●
●	○	●	×	○	○	○	●		○	○	●
○	●	●	○	×		●	●		●	○	○
●		○	○		×	○	●	●	○	●	●
●	●	●	○	●	○	×		●	●	●	
●	○	○	●	●	●		×	●			●
○	○	●			●	●	●	×		●	●
●	○	○	○	●	○	●			×		
●	●	●	○	○	●	●		●		×	●
○		●	●	○	●		●	●		●	×

Újraszínezzük az élekkel bővített gráf telítetlen pontjait:

1	2	3	4	5	6
2	4	5	7	10	12
6	9		8	11	

Nincs több nagyméretű színosztály, így a bővített gráf komplementeréből felírhatjuk a kapott feltételrendszer mellé a minimalizálandó kvadratikus célfüggvényt:

$$q(\mathbf{x}) = \mathbf{x}^{\mathrm{T}}\left(\overline{\mathbf{A}} - 2\mathbf{I}\right)\mathbf{x} = (x_2x_6 + x_2x_{12} + x_4x_9 + x_5x_6 + x_5x_9 + x_7x_8 + \\ + x_7x_{12} + x_8x_{10} + x_8x_{11} + x_9x_{10} + x_{10}x_{11} + x_{10}x_{12}) - \sum_{i=1}^{12} x_i^2$$

Megjegyzések

A fenti példában mohó színezéssel és kombinatorikus vágásokkal vontuk össze a feltételeket. Természetesen tetszőleges más módszerrel is tovább redukálható a feltételek száma vagy a célfüggvény mérete (általában egyik a másik kárára).

A kapott kvadratikus optimalizálási feladathoz ellenőrzés végett hozzávehető még az eredeti célfüggvény is, amellyel viszont már többcélú optimalizálási problémához jutunk.

Klikk-keresési problémák méretkorlátos, szimmetrikus átfogalmazása

Probléma

Keressük meg a legnagyobb méretű klikket az alábbi adjacencia-mátrixszal leírható gráfban:

×		•	•		•	•	•		•	•	
	×			•		•				•	
•		×	•	•		•		•		•	•
•		•	×				•				•
	•	•		×		•	•		•		
•					×		•	•		•	•
•	•	•		•		×		•	•	•	
•			•	•	•		×	•			•
		•			•	•	•	×		•	•
•				•		•			×		
•	•	•			•	•		•		×	•
		•	•		•		•	•		•	×

Méretkorlátos feltételrendszer kialakítása

A komplementer-gráf adjacencia-mátrixa sorainak összegére írjunk fel egyenlőtlenségeket úgy, hogy a főátlóban feltüntetjük az adott csomópont (komplementer gráfbeli) fokszámát, és az egyenlőtlenségek jobb oldalára ugyanezt a fokszámot írjuk:

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	
4	1			1				1			1	≤ 4
1	8	1	1		1		1	1	1		1	≤ 8
	1	4			1		1		1			≤ 4
	1		7	1	1	1		1	1	1		≤ 7
1			1	6	1			1		1	1	≤ 6
	1	1	1	1	6	1			1			≤ 6
			1		1	4	1				1	≤ 4
	1	1				1	5		1	1		≤ 5
1	1		1	1				5	1			≤ 5
	1	1	1		1		1	1	8	1	1	≤ 8
			1	1			1		1	4		≤ 4
1	1			1		1			1		5	≤ 5

Könnyen belátható, hogy bármely klikk az eredeti gráfban megoldása lesz ennek az egyenlőtlenségrendszernek. Ugyanis ha egy csomópont eleme a klikknek, akkor a saját sorának egyenlőtlenségében részt vesz a főátlóbeli elem, ám a többi pozitív együtthatójú elem definíció szerint nincs összekötve a csomóponttal, tehát nem lehet a klikkben; így egy klikk elemeinek sorának álló feltételek mind élesek. Ugyanakkor azon csomópontok feltételeiben, amelyek nem elemei a klikknek, eleve nem vesz részt a főátlóbeli elem, így az egyenlőtlenség triviálisan teljesül.

Vegyük észre, hogy a fenti egyenlőtlenségek szorosabbá tehetők úgy, hogy a fokszámok helyére az adott sornak megfelelő csomópont nem-szomszédjai között lehetséges legnagyobb klikkméretet tüntetjük fel, amire hatékony felső korlátot ad pl. a mohó színezés. Az első sor esetében például x_1 nem-szomszédai rendre x_2, x_5, x_9 és x_{12}, amelyeket mohón színezve:

1	2
2	5
9	12

A mohó színezés azonnal 2-re szorítja le az x_1 nem-szomszédai között lehetséges klikkek legnagyobb méretét, így az első egyenlőtlenség főátlóbeli eleme és jobb oldala 2-re cserélhető. Minden sorra végrehajtva a mohó színezést az alábbi, az előzővel megegyező méretű, de szorosabb feltételrendszert kapjuk:

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	
2	1			1				1			1	≤ 2
1	5	1	1		1		1	1	1		1	≤ 5
	1	2			1		1		1			≤ 2
	1		4	1	1	1		1	1	1		≤ 4
1			1	4	1			1		1	1	≤ 4
	1	1	1	1	3	1			1			≤ 3
			1		1	3	1				1	≤ 3
	1	1				1	3		1	1		≤ 3
1	1		1	1				3	1			≤ 3
	1	1	1		1		1	1	4	1	1	≤ 4
			1	1			1		1	2		≤ 2
1	1			1		1			1		4	≤ 4

Ha a klikk-méret felső korlátjának becslése helyett ténylegesen meghatározzuk soronként a maximális klikk méretét, akkor természetesen a lehető legszorosabb feltételrendszert állítjuk elő.

Klikk-keresés prekondicionálása dominancia-vizsgálatokkal

Dominancia, közvetlen dominancia, elsőrendű dominancia

Az mondjuk, hogy egy gráf nem szomszédos a és b csúcspontja közül az a (**közvetlenül**) **dominálja** a b-t akkor és csak akkor, ha b minden szomszédja a-nak is szomszédja: $N(b) \subseteq N(a)$

Állítás

Ha egy Γ gráf a csúcspontja dominálja a b csúcspontot, akkor bármely $\Delta \subseteq \Gamma$ k-klikk esetén létezik egy $\Delta' \subseteq \Gamma \setminus \{b\}$ k-klikk is.

Bizonyítás

Az állítás egyszerűen azon alapul, hogy ha a b csúcspont eleme a Δ klikknek, akkor a közvetlen dominancia miatt helyettesíthető a-val, máskülönben pedig $\Delta = \Delta'$.

Következmény

Ha egy gráfban maximális méretű klikket keresünk, akkor a dominált pontok a keresésből kihagyhatók.

Megjegyzés

Egy dominált pont törlése a gráfból nincs hatással az egyéb dominancia-viszonyokra.

Elsőrendű dominancia vizsgálat

A dominancia-viszonyok felderítésének hatékony algoritmusa van, amely azon alapul, hogy egy csúcspont által dominált pontok éppen a csúcspont nem-szomszédai által kifeszített részgráf izolált pontjai. Sorra véve tehát a gráf pontjait, első lépésben ki kell válogatni a soron következő csúcspont nem-szomszédait, majd megkeresni közöttük az izolált pontokat.

Példa

Tárjuk fel a dominancia-viszonyokat az alábbi adjacencia-mátrixszal leírható gráfban:

×		•	•		•	•	•		•	•	
	×			•		•				•	
•		×	•	•		•		•		•	•
•		•	×				•				•
	•	•		×		•	•		•		
•					×		•	•		•	•
•	•	•		•		×		•	•	•	
•			•	•	•		×	•			•
		•			•	•	•	×		•	•
•				•		•			×		
•	•	•			•	•		•		×	•
		•	•		•		•	•		•	×

Az 1. csúcs nem-szomszédai: 2, 5, 9, 12 – közöttük nincs izolált pont, így az 1. csúcs egyetlen más csúcspontot sem dominál közvetlenül.

A 2. csúcs nem-szomszédai: 1, 3, 4, 6, 8, 9, 10, 12 – közöttük nincs izolált pont, így az 2. csúcs egyetlen más csúcspontot sem dominál közvetlenül.

A 3. csúcs nem-szomszédai: 2, 6, 8, 10 – közöttük a 2 és a 10 csúcsok izoláltak, így ezeket a 3. csúcs közvetlenül dominálja.

…

A 9. csúcs nem-szomszédai: 1, 2, 4, 5, 10 – közöttük az 5 izolált pont, így azt a 9. csúcs közvetlenül dominálja.

stb.

Közvetett dominancia, másodrendű dominancia, megszorított dominancia

Az mondjuk, hogy egy gráf nem szomszédos a és b csúcspontja közül az a **dominálja** a b-t egy közös u szomszédra megszorítva akkor és csak akkor, ha b és u minden közös szomszédja a-nak és u-nak is közös szomszédja: $N(b) \cap N(u) \subseteq N(a) \cap N(u)$

Állítás

Ha egy Γ gráf a csúcspontja dominálja a b csúcspontot az u-ra megszorítva, akkor bármely $\Delta \subseteq \Gamma$ k-klikk esetén létezik olyan k-klikk is a Γ-ban, amely nem tartalmazza a (b,u) élt.

Bizonyítás

Az állítás egyszerűen azon alapul, hogy ha a (b,u) él eleme a Δ klikknek, akkor a közvetett dominancia miatt a b csúcs helyettesíthető a-val, máskülönben pedig $\Delta = \Delta'$.

Következmény

Ha egy gráfban maximális méretű klikket keresünk, akkor a (b,u) típusú élek a keresés során törölhetők, ami egy b és u közötti közvetlen dominanciát eredményezhet.

Példa

Előző példánk gráfjában a 9. csúcs dominálná az 1. pontot, ha nem létezne az $(1,4)$ él. Ámde a 12. csúcs dominálja az 1. pontot a 4. csúcsra megszorítva, ezáltal a maximális klikk-keresésből az $(1,4)$ él törölhető, ami által a 9. csúcs közvetlenül dominálja az 1. és a 4. csúcsot is, így a klikk-keresésből az $(1,4)$ él törlése után mindkét csúcspont is azonnal törölhetővé válik.

Bináris fák

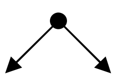
A bináris fákat többek között feladatok méretezésére, számítási igény megbecsülésére, illetve megoldás létezésének cáfolására használjuk.

Bináris fa

Egy irányított gráf bináris fa, ha

- irányítás nélkül fa;
- minden csúcsának befoka 0 vagy 1;
- és minden csúcsának kifoka 0 vagy 2.

Megjegyzések

Egy irányított fa  panelekből áll.

Egy bináris fának mindig páros sok éle van és páratlan sok csúcsa van.

Gyökérpont

Egy irányított fa gyökérpontja az a csúcsa, amelynek a befoka 0.

Megjegyzés

Egy irányított fának egyetlen gyökérpontja van. Ugyanis egy n csúcsú fának pontosan $(n-1)$ éle van, amelyeket irányítva $(n-1)$ befokhoz jutunk, így lesz egy olyan csúcs, amibe nem fut be él.

Végpont

Egy irányított fa végpontja az a csúcsa, amelynek a kifoka 0.

Megjegyzés

Egy irányított fának több végpontja is lehet.

Tétel

Egy bináris fának a gyökérpontjából minden pontba vezet egy egyértelmű irányított út.

Bizonyítás

Irányítás nélkül a fa bármely két pontja között egyértelmű út vezet, így a gyökérpont és bármely más pont között is. Irányítást a gyökérből kiindulva pedig csak egyféleképpen lehet a fára alkalmazni.

Pont szintje

Bináris fa egy pontjának szintje a gyökérből a ponthoz vezető út hossza.

Teljes bináris fa

Egy bináris fa teljes, ha az összes végpontja azonos szinten van.

Tétel

Egy n szintű teljes bináris fának $(2^n - 1)$ csúcsa van.

Bizonyítás

A csúcsokat szintenként összegezve a $2^0 + 2^1 + \ldots + 2^{n-1}$ mértani sort kapjuk.

Stratégia

Egy $(n+1)$ szintű bináris fa stratégiája egy n dimenziós pozitív egész számokból álló vektor, amelynek i-edik komponense megadja a bináris fa i szintű pontjainak a számát.

Megjegyzések

Az 1 pontú bináris fának nincsen stratégiája.

Egy több pontú bináris fának mindig egyértelmű stratégiája van.

A stratégia általában nem határozza meg egyértelműen a bináris fát. Azonos stratégiájú bináris fák általában nem izomorfak.

Egy tetszőlegesen választott n dimenziós egész komponensű vektor nem feltétlenül lesz stratégiája egy bináris fának.

Tétel

A $\sum_{k=1}^{n} d_k \cdot 2^{n-k} = 2^n$ tulajdonsággal rendelkező **d** vektorhoz létezik **d** stratégiájú bináris fa.

Tétel

Egy **d** stratégiájú bináris fa csúcsainak a száma: $\left(2 \cdot \sum_{k=1}^{n} d_k\right) - 1$

Bizonyítás

Gondolatban tegyük teljessé a bináris fát, majd vonjuk ki a fából a ténylegesen hiányzó csúcsokat:

$$\underbrace{\left(2^n - 1\right)}_{\text{teljes fa}} - \underbrace{\sum_{k=1}^{n} d_k \cdot \left(2^{n-k} - 2\right)}_{\text{hiányzó csúcsok}} = \underbrace{2^n - \left(\sum_{k=1}^{n} d_k \cdot 2^{n-k}\right)}_{=0} - 1 + \left(2 \cdot \sum_{k=1}^{n} d_k\right)$$

Megjegyzés

A stratégiából tehát egyfelől meg lehet becsülni egy bináris fa méretét, másfelől, ha csak a fa méretére vagyunk kíváncsiak, akkor nem fontos a teljes stratégiavektort tárolni: elegendő azt akkumulálni.

Feszítőfák

Minimális súlyú feszítőfa

A probléma leírása

Hálózatnak nevezünk egy élsúlyozott összefüggő gráfot.

Adott hálózaton keressünk minimális súlyú feszítőfát.

Megoldási módszerek

Kruskal algoritmusa

Az éleket súlyuk szerint növekvő sorrendben adjuk hozzá egy körmentes élhalmazhoz.

1. Rendezzük az éleket súly szerint növekvő sorrendbe.

2. Induljunk ki egy üres, tehát körmentes élhalmazból.

3. Bővítsük az élhalmazt a soron következő éllel, ha az körmentes marad.

Prim algoritmus

Tetszőleges csúcspontból kiindulva növesszünk fát, rendre az elérhető legkisebb súlyú él hozzáadásával.

1. Induljunk ki a gráf egy tetszőleges csúcsából, mint egypontú fából.

2. Bővítsük a fát az elérhető legkisebb súlyú éllel.

Folyamok

Minimális költségű folyam

Egy súlyozott élű irányított gráf csúcsaihoz kapacitási értékeket rendelünk úgy, hogy a kapacitások összege 0 legyen. Keressük meg a legkisebb költségű hálózati folyamot a gráfban!

A problémát szokás úgy interpretálni, hogy utakkal összekötött városokban valamely termék iránt kereslet vagy kínálat mutatkozik. Feltételezzük, hogy a teljes kereslet egyenlő a teljes kínálattal. Az utakhoz rendelt szállítási költségek alapján a cél a kereslet kielégítése minimális költséggel.

A gráf minden csomópontjának egy sor, minden élének egy oszlop felel meg a gráf illeszkedési mátrixában úgy, hogy a sorvektorok összefüggő rendszert alkotnak: a mátrix rangja éppen eggyel kevesebb, mint a csomópontok száma.

Az MKF problémával ekvivalens LP-probléma megfogalmazása

Változók bevezetése

Sorszámozzuk meg a gráf csomópontjait. A csomópontokhoz rendelt kapacitásokat jelölje b_k.

Az i-edik csomópontból kiinduló és a j-edik csomópontba érkező irányított élhez rendelt költséget vagy súlyt jelölje c_{ij}.

Az i-edik csomópontból kiinduló és a j-edik csomópontba érkező irányított élhez rendelt folyamot jelölje $x_{ij} \geq 0$.

Feltételek felírása

Írjuk fel a csomópontokat érintő folyamok mérlegegyenleteit:

$$\sum_{j=k} x_{ij} - \sum_{i=k} x_{ij} = b_k$$

Célfüggvény

Írjuk fel a folyamok és a hozzájuk tartozó súlyok szorzatösszegét.

$$z = \sum c_{ij} x_{ij}$$

Ennek a függvénynek keressük a minimumát.

Optimalitásvizsgálat

Lehetséges bázismegoldások

Az MKF probléma egy lehetséges bázismegoldása a hálózati gráfnak egy feszítőfáját határozza meg.

A duál feladat

Az MKF problémával ekvivalens LP-feladat duálja:

$$\begin{cases} \max w = \sum b_i y_i \\ y_i - y_j \leq c_{ij} \\ y_i \in \mathbf{R} \end{cases}$$

A duál feladat úgy interpretálható, hogy

Komplementaritási tétel az MKF problémára

A standard LP-problémára vonatkozó komplementaritási tétel alapján az MKF problémára megfogalmazott ekvivalens primál LP-feladat $\mathbf{x} = (x_{ij})$ megoldása és a duál feladat $\mathbf{y} = (y_k)$ megoldása akkor és csak akkor optimálisak, ha $x_{ij} > 0$ esetén $y_i - y_j = c_{ij}$.

Az $\mathbf{x}$ lehetséges megoldás akkor és csak akkor optimális, ha a *nembázis-változókhoz tartozó éleken* $\bar{c}_{ij} = y_i - y_j - c_{ij} \leq 0$.

Hálózati szimplex algoritmus

1. Alapelvünk, hogy nem szállítunk kétféle úton: induljunk ki tehát egy feszítőfa mentén meghatározott tetszőleges folyamból mint bázismegoldásból.

2. Rendeljünk $y_i = 0$ költségszintet a gráf egy tetszőleges pontjához, majd abból kiindulva a feszítőfa mentén számítsuk ki a többi ponthoz tartozó költségszintet úgy, hogy az irányítás mentén mozogva levonjuk, az irányítással szembe mozogva pedig hozzáadjuk az adott él költségét a végpont költségszintjéhez. $y_i - y_j = c_{ij}$

3. Számítsuk ki a $\bar{c}_{ij} = y_i - y_j - c_{ij}$ redukált költségszint-különbségeket azokon az éleken, amelyek nem tartoznak a feszítőfához. Ha ezek a redukált költségszint-különbségek sehol sem pozitívak, akkor a folyam optimális.

4. Ha találunk olyan élt, amelyen a redukált költségszint-különbség pozitív, akkor azt be kell venni a megoldásba úgy, hogy

- az élt csatoljuk a feszítőfához, amiben ekkor egy kör keletkezik;

- az új élhez rendelt folyam nagyságát a lehető legnagyobbra növeljük, hozzá igazítva a kör többi élfolyamának értékét;

- a folyamok átrendezésének eredményeként lesz a körön egy él, amelyen a folyam 0-ra csökken: ezt elhagyjuk a megoldásból.

5. Visszatérve a 2. lépéshez újra ellenőrizzük a megoldás optimalitását, egészen addig, amíg el nem érjük az optimális bázist.

Az MKF probléma speciális esetei

Az általános MKF problémát a következő megszorításokkal (feltételekkel) bővíthetjük:

- Pozitív kapacitások között nem engedélyezünk folyamot (kínálati pontok nem szállíthatnak egymásnak).

- Negatív kapacitások között nem engedélyezünk folyamot (keresleti pontok nem szállíthatnak egymásnak).

- Nem engedélyezünk nulla kapacitású csomópontot.

Amennyiben mindhárom kiegészítéssel élünk, egy MKF probléma egy meghatározott típusához, a **szállítási feladathoz** jutunk, amelyet **disztribúciós módszerrel** oldunk meg.

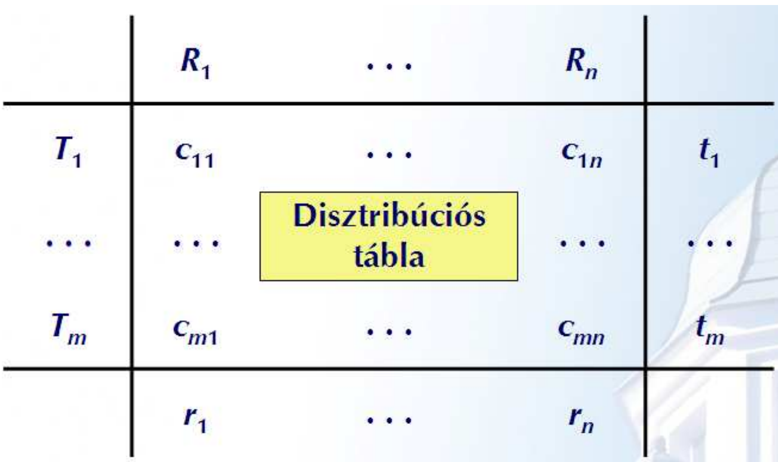
Szállítási feladat

A szállítási feladat egy irányított teljes páros gráffal modellezhető **MKF probléma**.

A szállítási feladatot szokás úgy megfogalmazni, hogy adott számú megrendelő meghatározott nagyságú termékigényét kell ismert számú és kapacitású telephelyről a lehető leggazdaságosabban kielégíteni. A telephelyek és a megrendelők közötti szállítási útvonalak meghatározott költségekkel terheltek. Szokás a tiltott útvonalakat a többihez képest jelentősen nagyobb (végtelen), a kötelező útvonalakat pedig a többihez képest jelentősen kisebb (nulla) költségekkel jellemezni.

Disztribúciós tábla

A szállítási feladatot – a teljes páros gráf reprezentáció mellett – egy disztribúciós táblával modellezhetjük, amelynek bal oldalán a telephelyeket, jobb oldalán azok kínálati kapacitását, felső részén a megrendelőket, alsó részén pedig azok keresleti igényeit tüntetjük fel. A disztribúciós tábla középső mátrixába a telephelyek és a megrendelő közötti szállítási költségek kerülnek.



Láthatjuk, hogy a disztribúciós tábla voltaképpen az irányított gráf súlyozott szomszédsági mátrixa:

	Megrendelők (*j*)	
Telephelyek (*i*)	Élköltségek (c_{ij})	**Kínálat (t_i)**
	Kereslet (r_j)	

A telephelyeket és a megrendelőket általában külön számozzuk, és mind a keresletet, mind a kínálatot pozitívan kezeljük.

Az ekvivalens LP-probléma és duálja

A kínálati feltételekhez rendelt duálváltozókat általában u_i-vel, a keresleti feltételekhez rendelt duálváltozókat általában v_j-vel jelöljük. Az **A** együtthatómátrix:

	x_{11}	x_{12}	x_{13}	x_{14}	x_{21}	x_{22}	x_{23}	x_{24}	x_{31}	x_{32}	x_{33}	x_{34}
u_1	1	1	1	1	0	0	0	0	0	0	0	0
u_2	0	0	0	0	1	1	1	1	0	0	0	0
u_3	0	0	0	0	0	0	0	0	1	1	1	1
v_1	1	0	0	0	1	0	0	0	1	0	0	0
v_2	0	1	0	0	0	1	0	0	0	1	0	0
v_3	0	0	1	0	0	0	1	0	0	0	1	0
v_4	0	0	0	1	0	0	0	1	0	0	0	1

Primál feladat

$$\begin{cases} \max z = \mathbf{cx} \\ \mathbf{A}_1\mathbf{x} = \mathbf{t} \\ \mathbf{A}_2\mathbf{x} = \mathbf{r} \\ \mathbf{x} \geq \mathbf{0} \end{cases}$$

Duál feladat

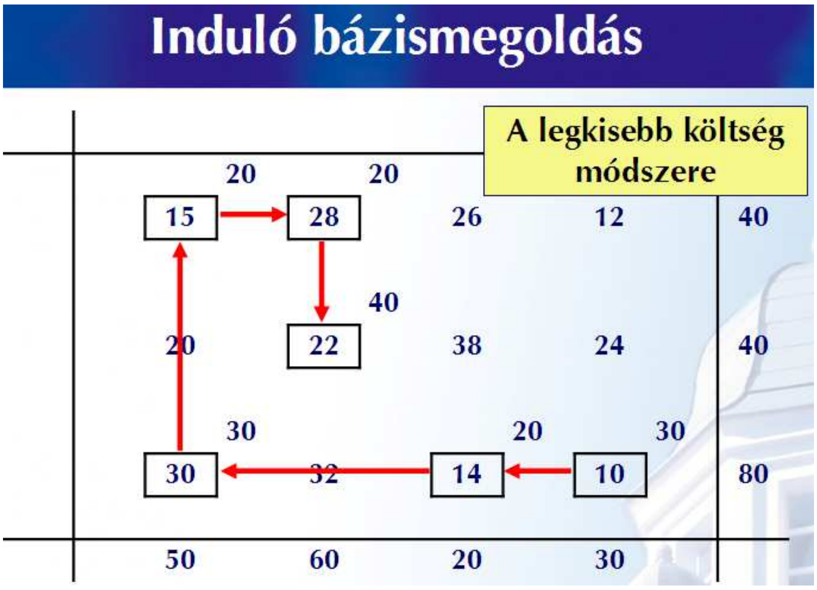
$$\begin{cases} \max w = \mathbf{ut} + \mathbf{vr} \\ \mathbf{uA}_1 + \mathbf{vA}_2 \leq \mathbf{c} \\ \mathbf{u} \in \mathbf{R}^m, \mathbf{u} \in \mathbf{R}^n \end{cases}$$

A legkisebb költség elvén alapuló disztribúciós módszer (*transportation simplex*)

A szállítási feladatot közvetlenül a disztribúciós táblán oldjuk meg:

1. Megjelöljük a legkisebb költségű szállítási útvonalat, és azt a hozzá tartozó telephely kapacitásától vagy a hozzá tartozó megrendelő igényétől függően maximálisan megterheljük. Ez azt jelenti, hogy ha az adott telephely kapacitásából futja, akkor az adott megrendelő teljes igényét ki kell elégíteni, máskülönben pedig a telephely teljes kapacitását el kell szállítani. Amikor egy x_{ij} folyamot rendelünk egy c_{ij} költségű élhez, akkor azt mondjuk: "lekötöttük a c_{ij} szállítást x_{ij} értékkel" (pl. "*Lekötöttük a 10-et 30-cal.*"). Az élekhez rendelt folyam nagyságát az él mellé írjuk.

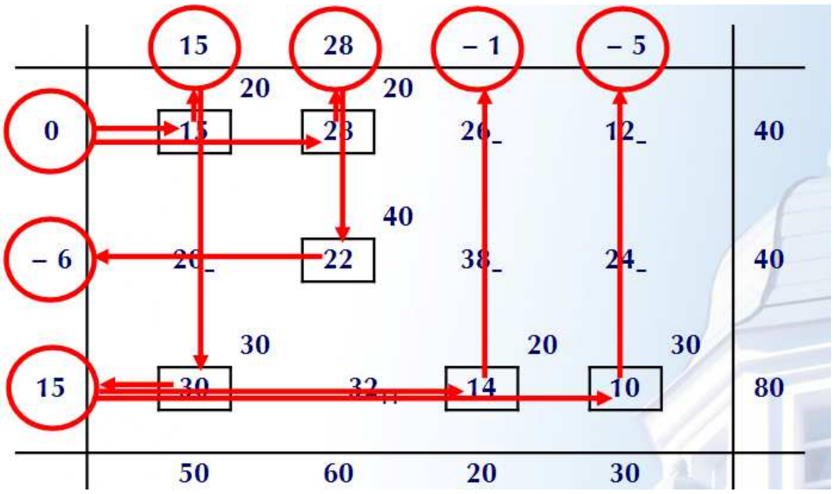
2. Amennyiben a megrendelő teljes igényét kielégítettük, úgy a telephely sorában, amennyiben a telephely teljes kapacitását kimerítettük, úgy a megrendelő oszlopában keressük meg a következő legkisebb költségű szállítási útvonalat, egészen addig, amíg minden kapacitást ki nem merítettünk és/vagy minden igényt ki nem elégítettünk. Az így kapott szállítási terv lesz az **induló bázismegoldás**:



3. A komplementaritási tétel alapján minden **x** bázismegoldásra $\mathbf{u}(\mathbf{t}-\mathbf{A}_1\mathbf{x})=0$ és $\mathbf{v}(\mathbf{r}-\mathbf{A}_2\mathbf{x})=0$ teljesül, duálisan tehát $(\mathbf{c}-\mathbf{u}\mathbf{A}_1-\mathbf{v}\mathbf{A}_1)\mathbf{x}=0$. Ebből fakadóan az **x** primál, illetve **u,v** duál bázismegoldások akkor és csak akkor optimálisak, ha $x_{ij}>0$ esetén $u_i+v_j=c_{ij}$.

Az **x** lehetséges megoldás akkor és csak akkor lesz tehát optimális, ha a *nembázis-változókhoz tartozó éleken* $\bar{c}_{ij}=u_i+v_j-c_{ij}\le 0$. Ezt a disztribúciós táblán ellenőrizni kell.

Rendeljünk 0 disztribúciós szintet egy tetszőleges duálváltozóhoz, majd abból kiindulva a bázismegoldás mentén számítsuk ki a többi duálváltozóhoz tartozó disztribúciós szintet a $u_i+v_j=c_{ij}$ összefüggés alapján:



Ezek után számítsuk ki és jelöljük a $\bar{c}_{ij} = u_i + v_j - c_{ij}$ kiegészítő szintek értékeit:

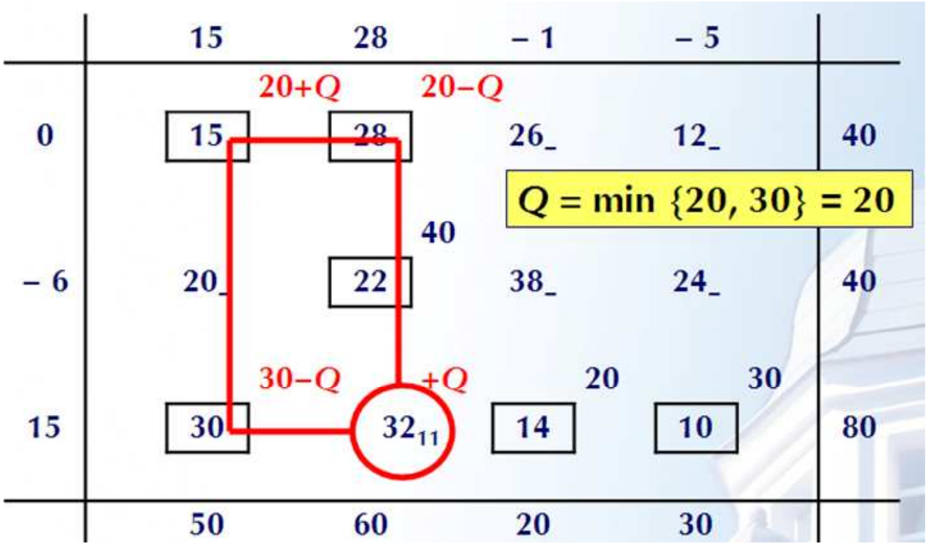
	15	28	− 1	− 5	
0	20 $\boxed{15}$	20 $\boxed{28}$	26_	12_	40
− 6	20_	40 $\boxed{22}$	38_	24_	40
15	30 $\boxed{30}$	32_{11}	20 $\boxed{14}$	30 $\boxed{10}$	80
	50	60	20	30	

4. Ha találunk olyan élt, amelyen a kiegészítő disztribúciós szint pozitív, akkor azt be kell venni a megoldásba úgy, hogy

- az élt csatoljuk a feszítőfához, amiben ekkor egy kör keletkezik;

- az új élhez rendelt folyam nagyságát a lehető legnagyobbra növeljük, hozzá igazítva a kör többi élfolyamának értékét;

- a folyamok átrendezésének eredményeként lesz a körön egy él, amelyen a folyam 0-ra csökken: ezt elhagyjuk a megoldásból.



5. Visszatérve a 2. lépéshez újra ellenőrizzük a megoldás optimalitását, egészen addig, amíg el nem érjük az optimális bázist.

Megjegyzés

Ha a $\bar{c}_{ij} = u_i + v_j - c_{ij}$ értékek között nincs pozitív, de van 0 értékű, akkor a hozzá tartozó útvonalat bevonva a megoldásba egy **alternatív optimumot** kapunk.

Nem klasszikus szállítási feladat

A szállítási feladat kis módosítása lehet, hogy **ha a kereslet és a kínálat eltér**: ilyenkor a felesleges többletet egy 0 költségű szállítási útvonalakkal rendelkező virtuális telephelyhez vagy megrendelőhöz rendeljük hozzá, és így futtatjuk a disztribúciós módszert.

Hozzárendelési feladat

A hozzárendelési feladat olyan **speciális szállítási feladat**, ahol minimális költségű teljes párosítást keresünk egy élsúlyozott páros gráfban.

A hozzárendelési feladattal ekvivalens LP-probléma

Változók bevezetése

Sorszámozzuk meg a páros gráf két csúcshalmazának elemeit külön-külön. Az *i*-edik és *j*-edik csúcsot összekötő él súlya legyen c_{ij} Az *i*-edik és *j*-edik csúcsot összekötő él kiválasztását a teljes párosításba jelölje $x_{ij} \in \{0,1\}$

Megjegyzés

A kiválasztási feltételeken szokás a következőképpen lazítani: $x_{ij} \in [0,1]$

Feltételek felírása

A teljes párosítás minden csúcsponthoz egyetlen kiválasztott élt rendel, így a teljes párosítás szomszédsági (adjacencia) mátrixának minden sorában és oszlopában pontosan egy 1-es szerepel:

- $\sum_j x_{ij} = 1$
- $\sum_i x_{ij} = 1$

Célfüggvény

Írjuk fel a teljes párosítás súlyát.

$$z = \sum c_{ij} x_{ij}$$

Ennek a függvénynek keressük a minimumát.

Magyar módszer

Hozzárendelési feladatra a disztribúciós módszer nem hatékony.

A magyar módszer a duál feladat megoldását célozza a költségmátrix redukciójával.

Duál feladat

$$\begin{cases} \max w = \sum_i u_i + \sum_j v_j \\ u_i + v_j \le c_{ij} \\ u_i, v_j \in \mathbf{R} \end{cases}$$

A költségmátrix redukciója

Vegyük észre, hogy a u_i illetve v_j duál változók értékei a $\mathbf{C} = (c_{ij})$ költségmátrixról közvetlenül leolvashatók: u_i rendre az *i*-edik sor legkisebb eleme, v_j pedig a *j*-edik oszlop legkisebb eleme!

Redukáljuk a költségmátrixot a következőképpen:

1. Csökkentsük minden sor elemeinek értékét a sorban található minimális elem (u_i) értékével.

2. Csökkentsük minden oszlop elemeinek értékét az oszlopban található minimális elem (v_j) értékével.

Komplementaritási tétel

A primál feladat **x** lehetséges megoldása és a duál feladat **u,v** lehetséges megoldása akkor és csak akkor optimális, ha $x_{ij} > 0$ esetén $u_i + v_j = c_{ij}$.

Vegyük észre, hogy a redukált költségmátrix elemei éppen a $c_{ij} - u_i - v_j$ értékek, a redukált költségmátrixban tehát az összes duál változó értéke 0.

Legyen $x_{ij} = 1$ ott, ahol $c_{ij} - u_i - v_j = 0$, mindenütt másutt pedig $x_{ij} = 0$.

A költségmátrix lefedése

1. Válasszuk ki a költségmátrixból a lehető legtöbb független 0 elemet.

Kőnig Dénes tétele, hogy egy páros gráfban a maximális párosítás éleinek a száma megegyezik a gráf éleit lefogó csúcspontok minimális számával.

Kőnig tétele alapján a redukált költségmátrixban a független nullák száma megegyezik az összes nullát lefedő sor- és oszlop-vonalak minimális számával.

2. Húzzuk meg a redukált költségmátrix 0 elemeit lefedő sor- és oszlopvonalakat.

A duál megoldás javítása

1. Keressük meg a legkisebb fedetlen elemet, legyen ez δ.

2. A fedetlen elemekből vonjuk ki δ-t.

3. Az egyszeresen fedett elemeket hagyjuk változatlanul.

4. A kétszeresen fedett elemekhez adjunk hozzá δ-t.

Megjegyzés

A fenti transzformációval ekvivalens megfogalmazás, hogy

1. A fedetlen sorok minden eleméből vonjunk ki δ-t.

2. A lefedett oszlopok minden eleméhez adjunk hozzá δ-t.

További ekvivalens megfogalmazást kaphatunk, ha felcseréljük a *sor* és *oszlop* szavakat.

Eredmény

Ha a lefedett sorok száma eredetileg I_s, a lefedett oszlopok száma pedig eredetileg I_o, akkor a fenti transzformációval a $w = \sum_i u_i + \sum_j v_j$ célfüggvény $(n - I_s)\delta$ mennyiséggel csökkent, és $I_o\delta$ mennyiséggel nőtt, összességében tehát a célfüggvény növekménye: $d = (n - I_s)\delta - I_o\delta = (n - I_s - I_o)\delta$

Amennyiben az eredeti párosításunk nem volt teljes, úgy $(n - I_s - I_o) > 0$, tehát a javítással a célfüggvény értéke nőtt.

Az eljárás vége

Ha a redukált, majd rekurzívan javított költségmátrixból kiválasztott maximális számú független 0 elemmel a mátrix minden sorát és oszlopát le tudjuk fedni, akkor az optimális megoldás a kiválasztott 0 elemeknek megfelelő éleket tartalmazó teljes párosítás.

Termeléstervezési (készletezési) feladat – Többperiódusú termeléstervezési probléma

Egy termék iránt meghatározott periódusonként változó kereslet mutatkozik, és az adott periódusokban a termék előállításának és raktározásának a költségei is különbözhetnek (a raktározási költség csak a periódus végén a készleten lévő termékek mennyiségétől függ). Készítsünk termelési programot adott számú periódusra nézve úgy, hogy a nyitó- és zárókészlet egyaránt 0, termelési és készletezési költségeink pedig minimálisak legyenek.

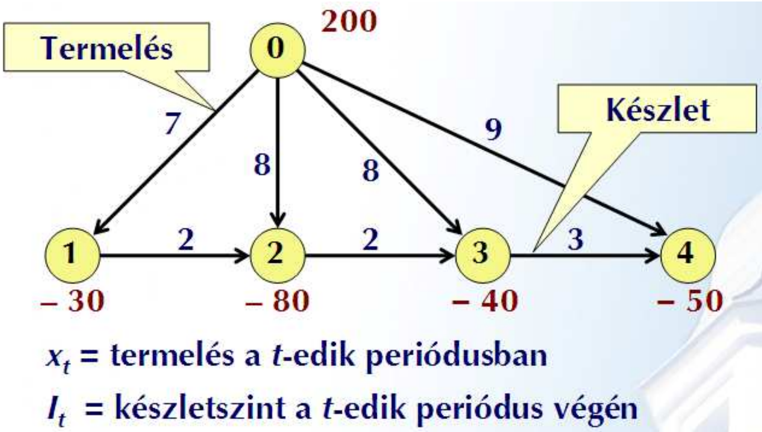
Az optimális termelési tervet egy **speciális MKF probléma** megoldása szolgáltatja.

Az ekvivalens MKF modell felállítása

Sorszámozzuk meg a periódusokat, és vegyünk fel egy 0 sorszámú termelési pontot. A termelési pont és a periódusokat jelentő pontok legyenek egy folyamgráf csúcsai. A termelési pontból minden egyes periódus ponthoz irányítsunk élt, és az élek súlya legyen rendre a megfelelő periódushoz rendelt termelési költség. Minden periódus pontból irányítsunk a rákövetkező periódus ponthoz élt, és az él súlya legyen rendre a kiinduló periódushoz rendelt készletezési költség.

Minden periódus ponthoz rendeljük hozzá negatív kapacitásként az adott periódusban a termék iránt mutatkozó keresletet. A termelési ponthoz rendeljünk hozzá pozitív kapacitásként a teljes kereslet kielégítéséhez szükséges összes termelési szükségletet.

Például:



Az ekvivalens LP-modell felállítása

Változók bevezetése

Jelölje b_t a t-edik periódusban a termék iránt mutatkozó keresletet.

Jelölje c_t a t-edik periódus termelési költségét.

Jelölje d_t a t-edik periódus készletezési költségét.

Jelölje x_t a t-edik periódusban termelt mennyiséget, azaz a termelési pontból a t-edik periódus ponthoz irányított folyamot.

Jelölje I_t a t-edik periódus végén készleten maradt mennyiséget, azaz a t-edik periódus pontból a (t+1)-edik periódus ponthoz irányított folyamot.

Feltételek felírása

Az egyes periódusok mérlegegyenletének egyik oldalán a nyitókészlet és a megtermelt mennyiség, a másik oldalán az eladások és a zárókészlet áll:

$$I_{t-1} + x_t = b_t + I_t$$

A teljes nyitó- és zárókészlet 0.

Célfüggvény

$$z = \sum_t \left(c_t x_t + d_t I_t \right) \rightarrow \min$$

Az ekvivalens szállítási feladat megfogalmazása

A termeléstervezési probléma szállítási feladatként is megfogalmazható úgy, hogy többszörös telephelyként a teljes keresleti potenciállal rendelkező termelési pontot tüntetjük fel, és az így keletkező kínálati többlet elnyeléséhez egy olyan fiktív megrendelőt vezetünk be, akihez 0 szállítási költséggel lehet a terméket eljuttatni.

Mivel a termeléstervezési probléma gráfja eredetileg nem teljes páros gráf, ezért a költségmátrix felírásakor üres pozíciók is keletkeznek a hiányzó élek miatt: ezeket mint tiltott éleket kezeljük, tehát egy ésszerűtlenül magas (végtelen) költséggel vesszük őket számításba.

Például:

	1	2	3	4		
1	7	9	11	14	0	200
2	*M*	8	10	13	0	200
3	*M*	*M*	8	11	0	200
4	*M*	*M*	*M*	9	0	200
	30	80	40	50	600	

Az optimális megoldás elemzése

A többperiódusú termeléstervezési problémának mindig van optimális feszítőfa megoldása. Az optimális termelési terv elve egészen egyszerűen fogalmazható meg: egy adott periódusban csak akkor termelünk, ha a nyitókészlet 0, és a termelés során mindig néhány előttünk álló periódus teljes keresletét megtermeljük. A magasabb előállítási (és alacsonyabb készletezési) költségű periódusok termékkészletét tehát a relatíve alacsonyabb termelési költségű periódusokban állítjuk elő.

Maximális folyam probléma

Számítsuk ki a maximális folyam értékét egy élkapacitásokkal ellátott, egy forrással és egy nyelővel rendelkező irányított hálózati gráfban!

Ebben a feladatban, ami egyébként a Ford-Fulkerson algoritmussal hatékonyan megoldható, a folyam áramlásának a költsége nem játszik szerepet.

A maximális folyam probléma azonban kis módosítással MKF problémaként is megfogalmazható!

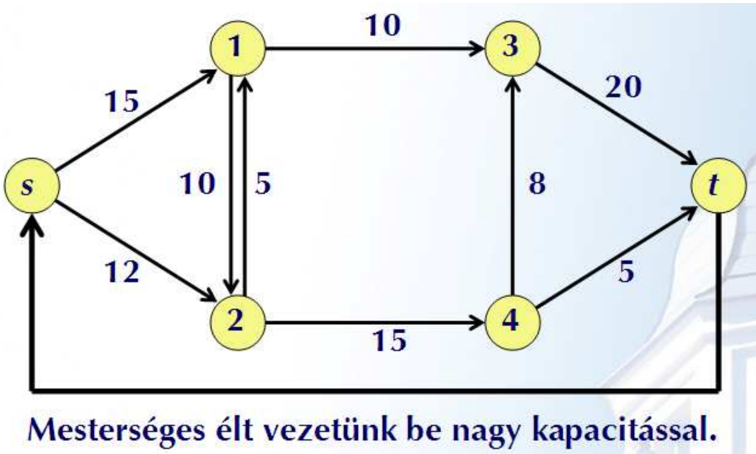
Az ekvivalens MKF modell felállítása

Rendeljünk minden eredeti élhez nulla szállítási költséget.

Vezessünk be egy a nyelőtől a forráshoz (vissza)irányított nagy kapacitású mesterséges élt, amelyhez 1 szállítási költséget rendelünk!

Keressük meg az így kialakított hálózati gráfban a minimális költségű folyamot.

Például:



Az ekvivalens LP-modell felállítása

Változók bevezetése

Sorszámozzuk meg a pontokat.

Az *i*-edik és *j*-edik csúcsot összekötő él kapacitása legyen $k_{ij} \geq 0$

Az *i*-edik és *j*-edik csúcsot összekötő él súlya (szállítási költsége) legyen $c_{ij} \geq 0$

Az *i*-edik és *j*-edik csúcsot összekötő élen áramló folyam nagysága legyen $x_{ij} \geq 0$

Feltételek felállítása

$$k_{n0} = \infty$$

$$c_{n0} = 1, c_{ij} = 0$$

A befolyó és kifolyó áramokra vonatkozó mérlegegyenletek:

$$\sum_i x_{ik} = \sum_j x_{kj}$$

Célfüggvény

$$z = x_{n0} \rightarrow \max$$

Utak

Legrövidebb út probléma

Határozzuk meg egy élhosszokkal ellátott gráf valamelyik (vagy összes) pontjából a gráf többi pontjába vezető legrövidebb utat és annak hosszát.

Ez a feladat a Dijkstra algoritmussal hatékonyan megoldható, de irányított gráf esetén létezik vele ekvivalens MKF, szállítási feladat, illetve hozzárendelési feladat is!

Az ekvivalens MKF modell felállítása

Sorszámozzuk meg a pontokat legyen a kiindulópont kínálata $n-1$, a többi pont kereslete 1.

Az ekvivalens szállítási feladat megfogalmazása

Sorszámozzuk meg a pontokat legyen a kiindulópont kínálata 1, a célpont kereslete 1, és a többi csúcspont csak átszállítási pont.

Ez azt jelenti, hogy a disztribúciós tábla felállításánál minden kínálat és kereslet értéke 1 lesz, és a teljes páros gráfhoz hiányzó éleket tiltottnak (végtelen költségűnek) jelöljük.

Az ekvivalens hozzárendelési feladat megfogalmazása

Az ekvivalens szállítási feladat disztribúciós táblája voltaképpen egy hozzárendelési feladat költségmátrixának is tekinthető, és azon a probléma a magyar módszerrel megoldható.

Az utazó ügynök problémája (*Travelling salesman's problem*)

Probléma

Egy ügynöknek egy körútján meghatározott településeket kell felkeresnie. Az utazás során egy települést csak egyszer kell/szabad érintenie. Ismert a települések távolsága, illetve a közöttük való utazás költsége, amelyet az ügynök minimalizálni akar. Keressük meg a települések bejárási sorrendjét, amely a legkisebb utazási összköltséggel jár.

Modell

Sorszámozzuk (indexeljük) meg a településeket. A településhálózat térképe egy teljes, élsúlyozott, irányított gráf lesz. Ezen a gráfon kell egy adott pontból a legkisebb költségű Hamilton-kört megkeresni.

Vezessük be a következő, bináris változókat:

$$x_{ijk} = \begin{cases} 1, \text{ ha a } k \text{ lépésben az } i \text{ pontból a } j \text{ pontba utazunk} \\ 0, \text{ egyébként} \end{cases}$$

A $\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{i=1}^{n} d_{ij} \cdot x_{ijk}$ célfüggvényt minimalizálni kell.

A feltételek a következőképpen alakulnak:

- Bármely településről pontosan egyszer távozunk: $\forall i : \sum_{j,k} x_{ijk} = 1$
- Bármely településre pontosan egyszer érkezünk: $\forall j : \sum_{i,k} x_{ijk} = 1$
- Bármely lépésben meghatározott, hogy honnan hová utazunk: $\forall k : \sum_{i,j} x_{ijk} = 1$
- Az egymást követő lépések összekapcsolódnak: $\forall j,k : \sum_{i=1}^{n} x_{ijk} = \sum_{v=1}^{n} x_{jv(k+1)}$
- Az utazás végén hazatérünk: $\forall j : \sum_{i=1}^{n} x_{ijn} = \sum_{v=1}^{n} x_{jv1}$

A probléma mérete a következőképpen alakul:

- a változók száma: n^3
- a feltételek száma: $n^2 + 3n$

A gyakorlatban a változók száma *n*-nel csökkenthető, ha elhagyjuk az $x_{iik} = 0$ alakú változókat.

Az összefüggőségi feltételek száma szintén csökkenthető, ha előre rögzítjük, hogy melyik településről indulunk, így

- $\forall j : \sum_{i=1}^{n} x_{ij1} = \sum_{v=1}^{n} x_{jv2}$ helyett $\forall j : x_{1j1} = \sum_{v=1}^{n} x_{jv2}$ írható;
- és $\forall j : \sum_{i=1}^{n} x_{ijn} = \sum_{v=1}^{n} x_{jv1}$ helyett $\forall j : \sum_{i=1}^{n} x_{ijn} = x_{j11}$ írható.